Master's thesis
Computational materials physics

Accelerated Structure Prediction of Halide Perovskites with Machine Learning

Jarno Laakso

March 12, 2021

Supervisors: Dr. Milica Todorović, Aalto University
Prof. Patrick Rinke, Aalto University

Examiners: Prof. Patrick Rinke, Aalto University
Prof. Kai Nordlund, University of Helsinki

UNIVERSITY OF HELSINKI
Master's Programme in Materials Research

P.O. Box 64 (Gustaf Hällströmin katu 2)
FI-00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Education programme
Faculty of Science	Master's Programme in Materials Research

Tekijä — Författare — Author
Jarno Laakso

Työn nimi — Arbetets titel — Title
Accelerated Structure Prediction of Halide Perovskites with Machine Learning

Opintosuunta — Studieinriktning — Study track
Computational materials physics

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	March 12, 2021	64

Tiivistelmä — Referat — Abstract

Halide perovskites are a promising materials class for solar energy production. The photovoltaic efficiency of halide perovskites is remarkable but their toxicity and instability have prevented commercialization. These problems could be addressed through compositional engineering in the halide perovskite materials space but the number of different materials that would need to be considered is too large for conventional experimental and computational methods. Machine learning can be used to accelerate computations to the level that is required for this task.

In this thesis I present a machine learning approach for compositional exploration and apply it to the composite halide perovskite $CsPb(Cl, Br)_3$. I used data from density functional theory (DFT) calculations to train a machine learning model based on kernel ridge regression with the many-body tensor representation for the atomic structure. The trained model was then applied to predict the decomposition energies of $CsPb(Cl, Br)_3$ materials from their atomic structure. The main part of my work was to derive and implement gradients for the machine learning model to facilitate efficient structure optimization.

I tested the machine learning model by comparing its decomposition energy predictions to DFT calculations. The prediction accuracy was under 0.12 meV per atom and the prediction time was five orders of magnitude faster than DFT. I also used the model to optimize $CsPb(Cl, Br)_3$ structures. Reasonable structures were obtained, but the accuracy was qualitative. Analysis on the results of the structural optimizations exposed shortcomings in the approach, providing important insight for future improvements. Overall, this project makes a successful step towards the discovery of novel perovskite materials with designer properties for future solar cell applications.

Avainsanat — Nyckelord — Keywords
Materials physics, Machine learning, Perovskite solar cells

Säilytyspaikka — Förvaringsställe — Where deposited

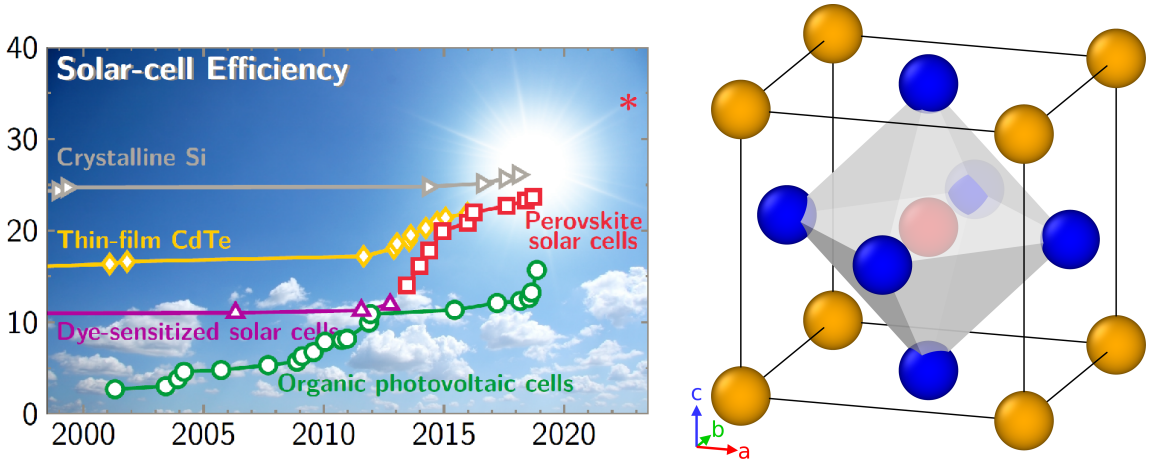
Muita tietoja — övriga uppgifter — Additional information

Contents

1. Introduction

The use of fossil fuels is the primary driving factor behind the green house effect [1]. Replacing them with renewable energy sources is a pressing challenge in the fight against global warming. During the past decade, the use of solar and wind power has increased rapidly, but even both of them together have not been able to turn the tide under the pressure of increasing energy demands [2]. Improved renewable energy generation is needed to accelerate the adoption of green technologies and mitigate climate change.

The sun provides ample energy, but solar energy generation has been held back from wide scale adoption due to its high price and low efficiency. Addressing these challenges requires both engineering and materials solutions. On the materials side, hybrid perovskites are offering a promising path towards more efficient solar cells (see Figure 1.1a). The hybrid perovskite $CH_3NH_3PbI_3$ ($MAPbI_3$) has triggered this renewed interest into new photovoltaic materials. Since its promising photovoltaic properties were discovered just a decade ago, $MAPbI_3$ solar cells have reached power conversion efficiencies of up to 25% in the lab [3]. However, the toxicity of lead and instability of $MAPbI_3$ in air and moist environments have hampered commercialization of this new technology [4, 5, 6]. These problems, however, could be addressed through material design [7, 8]. The materials design objective is to optimize the materials properties by changing the structure and composition in the hybrid perovskite materials space. This is not a trivial task, as many combinations need to be tried



(a) (b)

Figure 1.1: (a) Photovoltaic power conversion effictiency progression of different material families [9]. (b) Perovskite unit cell. A-site cations are colored yellow, B-site cation is red, and X-site anions are blue.

and tested. Conventional experimental and computational methods soon reach their limits leaving large parts of the design space unexplored. Machine learning might be the facilitator that builds on experimental or computational data to unlock a larger design space and to discover new materials to address the photovoltaic materials challenge.

1.1 Perovskites

1.1.1 Structure

The elemental formula of basic perovskites is ABX_3 where A and B are cations and X is an anion. At its simplest, the perovskite structure has cubic symmetry. One way to depict the atomic positions is to set the B cation in the center of the cubic cell. In this setting the A cations are located at the corners of the cell and the X anions lay at the centers of the six faces of the cube. This depiction is visualized in Figure 1.1b. The image also shows the BX_6 coordination octahedron that encapsulates the

B anion in its center.

In the ideal cubic structure, the coordination numbers of the A and B cations are 12 and 6, respectively. Often the coordination numbers are reduced, which breaks the cubic symmetry and manifests as tilting of the BX_6 octahedra. The atomic positions in the structure shift so that the shape of the octahedra remains but their orientation changes. A notation for classifying octahedral tilting in perovskites was introduced by Glazer [10]. In this notation the tilting system is defined by three rotational angles

$$(R_a^{\pm} R_b^{\pm} R_c^{\pm}),$$

where, R_a, R_b, and R_c are the rotation angles of the octahedra around the three lattice vectors. The superscripts + and − correspond to in-phase and antiphase tilting, respectively. In-phase tilting means that every octahedron layer is rotated to the same direction. Antiphase tilting means that following layers are rotated in opposite directions. If there is no tilting in some direction, it can be expressed with the superscript 0.

There are 23 Glazer tilting systems in total but regarding the rest of this thesis only four of them carry greater significance. The first of the four is the ideal cubic symmetric case with no tilting in any of the three directions ($R_a^0 R_b^0 R_c^0$). The space group of this system is $Pm\overline{3}m$. The next two systems are the 1-tilt systems ($R_a^0 R_b^0 R_c^+$) and ($R_a^0 R_b^0 R_c^-$). These structures have tetragonal lattices. The system with in-phase tilting corresponds to the space group P4/mbm, whereas the anti-phase system corresponds to I4/mcm. The fourth significant system is ($R_a^+ R_a^- R_c^-$). It has tilts in all three directions, making it the most complex of the four. It has an orthorhombic lattice structure and it corresponds to the space group Pnma.

The number of elemental combinations that can have the ABX_3 perovskite structure is large. From the photovoltaics perspective, halide perovskites is the most promising subcategory of perovskites. In them, the X-site anion is a halogen

(F^-, Cl^-, Br^-, I^-) and the B-site cation a divalent metal [11]. The possibilities for the A-site occupant vary greatly from alkali metals to small organic molecules, which is the case with $MAPbI_3$ where the A-site cation is methylammonium (CH_3NH_3).

The structural complexity rises even further for alloys. In alloys, the element on one site is partially substituted by another. $AB(XX')_3$ is an example of a binary alloy in which the anion site is divided between two elements: X and X'. The other perovskite sites can of course be alloyed, too, giving rise to complex compositions (ternary, quarternary, etc.). The complexity of alloys grows combinatorially with the number of substituents and dimensions, which leads to a severe exploration challenge, but unprecedented opportunities for materials design.

1.1.2 Properties

In this thesis I will focus only on halide perovskite materials for solar cell applications. Promising candidates for future solar energy materials have been discovered in both alkali metal and organometal halide perovskites. For example, power conversion efficiency (PCE) values of over 15% have been reported for $CsPbI_3$ [12]. Its organometallic counterpart $(CH_3NH_3)PbI_3$ ($MAPbI_3$) has shown even higher efficiency of $\sim 25\%$ [3]. The fundamental mechanisms behind the high PCE of halide perovskites remains unclear [13], but some beneficial properties are known. The binding energies between the photosynthesized excitons are small, aiding in the creation of charge carrying electrons and holes [14]. The diffusion lengths and lifetimes of these charge carriers have also been found to be large [15, 16]. Lastly, halide perovskites often have direct band gaps, which improves the photon absorption coefficient [17].

The attractiveness of halide perovskites as solar cell materials is further increased by the potential for low manufacturing costs. They can be synthesized at relatively low temperatures, which reduces the energy requirements during manufac-

turing. Furthermore, due to the high absorption coefficients only a thin perovskite layers are needed in devices, reducing the raw material requirements and increasing the potential for commercial viability.

Despite their beneficial properties, halide perovskites also face challenges. The best performing materials tend to contain lead (Pb) as their B-site cation, which makes them toxic. Considering the environmental implications, the large-scale adoption of such materials would be highly questionable. The second large hurdle is their low stability. They are soluble in water and degrade quickly when exposed to UV radiation, which is obviously a large problem in solar energy applications. Luckily, both the toxicity and the instability can be tackled with different design methods.

1.1.3 Design

To make halide perovskites viable for solar energy applications, a materials design problem needs to be solved. The problem has three criteria: The material should be stable, free of lead, and it should have an optimal band gap to guarantee high PCE. There are many methods in materials science that can help in the optimization of the properties of a material by controlling its structure. For halide perovskites, the most prevalent method is compositional engineering, where A, B, or X-site atoms of the perovskite are partly substituted with atoms of another element [18].

Elemental substitutions in $MAPbI_3$ have been studied extensively. The most common A-site substitution has been formamidinium ($HC(NH_2)_2$), which has been shown to improve stability without compromising on the PCE [19, 20]. Also inorganic A-site substitution has been attempted for example with Cs [21]. The effects on stability have been positive while the efficiency has suffered. To find lead-free halide perovskites, B-site substitutions have to be considered. Thus far, Sn, Bi, and Ge have been studied, but the PCE of the material has been negatively affected [8, 22]. X-site substations with elements such as Br affect the band gap of

$MAPbI_3$ [23], which means that it could potentially be used in tandem with other substitutions at other sites to obtain desirable results.

1.2 Research Approach

Even though compositional engineering has already been applied for halide perovskites, most of the studies have only looked at selected composition concentrations. To find the material composition where all the design criteria are met, we need a way to scan the entire composition range. It is possible that achieving the design criteria for $MAPbI_3$ will require simultaneous substitutions in all three atom sites, and that one site may need to have more than two elements. This means that the number of materials that need to be considered greatly surpasses what experimental or conventional methods can manage. A potential solution to the problem is machine learning, which can be used to speed up the existing computational methods considerably.

Even with the help of machine learning, solving the design problems of a complex material like $MAPbI_3$ is very difficult. For this reason, the problem was simplified both in terms of the material and optimized properties. The first simplification was to study an inorganic halide perovskite $CsPb(Cl, Br)_3$ instead of $MAPbI_3$ and all the possible substitutions that go with it. Focusing on a material that has no organic molecules decreases the degrees of freedom in the atomic structure. Additionally, the MA molecules in $MAPbI_3$ show long range ordering effects [24], which means that larger simulation systems would have to be used in order to fully characterize the material. Limiting the elemental substitutions to one site also simplifies the problem considerably. The second simplification was to only consider one of the three design criteria: stability.

Trying to find the Cl concentration that optimizes the stability of the perovskite $CsPb(Cl, Br)_3$ is a complicated problem in its own right. Even with 40 atom

simulation cells, a given concentration can correspond to over a million structures with different site permutations of the Cl and Br atoms. Furthermore, the stability of any given structure is not easy to asses. The equilibrium geometry depends on the elemental composition, which means that for every new permutation of atoms the structure needs to be optimized again. The stability of an equilibrium structure can be evaluated in terms of its decomposition energy, which is the amount of energy required to break the composite material into its building components. Both the structure optimization as well as the calculation of the decomposition energy can be done using conventional computational methods, such as density functional theory (DFT), but with such a large number of atomic structures, the computational costs would be intractable.

In this work, I developed a machine learning model based on DFT input data. Only a relatively small number of DFT calculations were performed for the $CsPb(Cl, Br)_3$ system. Then, I trained predictive machine learning model to learn the relationship between atomic geometry and decomposition energy. The trained machine learning model can make predictions for new geometries nearly instantly and thus much faster than DFT. I then implemented gradients for the machine learning model to facilitate efficient structure optimization.

The machine learning model that was selected for this work is a combination of the many-body tensor representation (MBTR) [25] and kernel ridge regression (KRR). MBTR is used to represent the atomic geometries in a vector form and KRR maps the vectors to decomposition energy values. Both components were selected partly because they are easy to differentiate, allowing to use the model for force calculations and structural optimization. Additionally, KRR works well with small data sets of less than 10 000 training instances [26], which should decrease the number of DFT calculations that are needed.

1.3 Objectives

The objective of this project was to develop and test a data-driven machine learning model that can be used in finding the Cl concentration that optimizes the stability of $CsPb(Cl, Br)_3$. The data sets for training and testing the model already existed, and I only analyzed them to make sure that they are appropriate. The remaining tasks are:

1. Use the data to train a predictive machine learning model
2. Derive and implement gradients for the machine learning model
3. Use the model for structural optimization

If the objectives are successfully completed, the benefits would go much further than stability optimization. The same methodology could be used to optimize other material properties. It could also be adopted in the study of more complex perovskites, such as $MAPbI_3$, and other compounds and alloys. Furthermore, structural optimization is a very common task in materials research. Accelerated optimization methodology using machine learning would benefit the whole field of study.

This thesis presents the steps that I took to complete the objectives. Chapter 2 explains the underlying methods that were used from DFT to the definition of the machine learning model. Chapter 3 introduces the data sets for training and testing the model. In Chapter 4 I fit the model to the data and test its capabilities in decomposition energy prediction. The model is differentiated in Chapter 5 where I also test the model in structural optimization. The thesis is concluded by Chapter 6 where I discuss the results of the work and make suggestions on the possible future improvements.

2. Methods

In the previous chapter, some computational methods used in the design of perovskite materials were mentioned. In this chapter, the emphasis will be on the methods that are relevant to the work presented in this thesis. The first of these methods is density-functional theory (DFT) [27, 28, 29, 30, 31, 32]. DFT is a quantum mechanical method used for calculating physical and chemical properties of atomic systems. It has become the method of choice for such purposes due to its versatility and high accuracy compared to computational costs.

The use of conventional computational methods, such as DFT, proves to be too computationally expensive for solving problems where many calculations are needed. This is why machine learning methods have recently emerged as a way to bridge the gap between the accurate but slow methods and complex research problems. The machine learning model that was used in this work is defined in the last section of the chapter.

2.1 Density Functional Theory

DFT is a computational method for computing the energy of an atomic system. It is based on the notion that the energy of an electronic system can be determined purely through the density of electrons [27]. In this chapter, I explain the theory behind the method and list some of its applications and advantages over alternative methods.

In quantum mechanics, the quantum states of isolated systems are described with wave functions Ψ. The operator for the total energy of the system is the Hamiltonian operator $\hat{H}$, and the energy of the system is its expectation value:

$$E = \langle \Psi | \hat{H} | \Psi \rangle \,. \tag{2.1}$$

Atomic systems consist of nuclei and electrons. Because of the large mass difference between the two, it is reasonable to consider the wave functions of the electrons and the nuclei separately. This is called the Born–Oppenheimer (BO) approximation. The BO Hamiltonian for the electron system is

$$\hat{H} = \hat{T} + \hat{V}_{ee} + \hat{V}_{ext}, \tag{2.2}$$

where $\hat{T}$ is the kinetic energy operator, $\hat{V}_{ee}$ describes the interactions between the electrons, and $\hat{V}_{ext}$ is the external potential created by the nuclei.

Even in the Born–Oppenheimer approximation, finding the energy from the many-electron wave function is almost always impossible in practice. The wave function is a function of all the electron coordinates $\boldsymbol{r}_i$:

$$\Psi = \Psi(\{\boldsymbol{r}_i\}). \tag{2.3}$$

In other words, the number of arguments in the wave function scales with the number of electrons. In most materials science problems, the systems of interest have at least tens of atoms and hundreds of electrons. Using the wave function description to find the energy of systems like this is not attainable analytically or numerically.

In DFT, the problem is avoided by switching to the electron density $n(\boldsymbol{r})$. Unlike the wave functions, $n(\boldsymbol{r})$ is always three dimensional regardless of the system size. The argument for why this is meaningful is given by the Hohenberg-Kohn theorems [27]. The first theorem states that the ground state electron density $n_0(\boldsymbol{r})$ describes uniquely the external potential V_{ext}, and thus the whole BO Hamiltonian. In other words, the properties of the system are determined by $n(\boldsymbol{r})$. The second

theorem states that the ground state density $n_0(\boldsymbol{r})$ always corresponds to the ground state energy E. Based on the theorems, there exists a universal functional $E[n]$ that determines the ground state energy of the electron system in any V_extt based on the ground state electron density $n_0(\boldsymbol{r})$.

What the Hohenberg-Kohn theorems do not reveal is the functional form of this universal functional $E[n]$ and how the ground state electron density n_0 can be found. The Kohn-Sham approach [33] is the most common way to solve this problem. The main idea of the approach is to treat the electron system as a system of non-interacting electrons in an external potential. The remaining interaction terms of the Hamiltonian are bundled together into so called exchange-correlation potential V_{xc}. The energy functional becomes

$$E[n] = T_0[n] + V_{ext}[n] + E_c[n] + E_{xc}[n], \tag{2.4}$$

where T_0 is the kinetic energy of non-interacting electrons, V_{ext} is the external potential, E_c is the classical Coulomb energy of the electrons, and E_{xc} is the energy related to V_{xc}. What is important here, is that the three first terms are known exactly and they can be evaluated given n. $E_{xc}[n]$ is not known exactly and needs to be approximated. The approximations that are in use today are numerous: from the simple local density approximation (LDA) [28], to more comprehensive general gradient approximations (GGA), such as the Perdew-Burke-Ernzerhof (PBE) functional [34], and beyond. The Kohn-Sham approach leads to a set of equations that can be used to iteratively solve for n_0 and the corresponding ground state energy.

On top of calculating ground state energies, DFT is also capable of much more. Many energy functionals can be differentiated with regards to the atomic positions, which allows using DFT for structural optimizations. Solving n_0 provides information of the electronic structure of the system, and thus DFT can be used to simulate different electronic and optical properties of materials.

DFT is now the most established method for computing atomic scale properties

that require quantum mechanical insight in physics, chemistry and materials science. It is widely available in efficient and well-tested computer codes. In this work, we use the DFT implementation in the FHI-aims code [35] (see Section 3.1.2).

2.2 Machine Learning in Materials Science

Machine learning (ML) is a class of computational methods that learn from data without explicit instructions [36, 37]. During the past couple of decades, the use of machine learning has increased rapidly due to the wide availability of data. Sophisticated machine learning methods have been developed for tasks that range from image recognition to language translation and decision making in self-driving cars. Many of these methods can also be applied to scientific research problems.

There are many problems in materials science that can benefit from machine learning. When studying complex materials, the number of different variations to be considered is too large for experimental studies. Conventional computational methods, such as DFT, can also be heavily limited by high computational costs. Supervised learning methods can be used to find mappings from the structure of a material to its properties, such as energy or band gap [26, 38, 39, 40]. Predicting the properties with a machine learning model can be orders of magnitude faster than the conventional computational methods, while only compromising minimally on the accuracy.

There are four steps that go into creating a data-driven machine learning model for property prediction:

1. Building a data set
2. Structure description
3. Machine learning method selection and training

4. Quality control

The data set consists of structural data and property labels. There are different kinds of structural data, but here I will only focus on atomic geometries. In materials science, a common source of labeled data are DFT calculations. The quality of the data is important because the predictions from a machine learning model can only be as accurate as the data that was used to train it.

The goal of structure description is to represent the structural data in a form that is compatible with the machine learning model. Most machine learning methods work on data that is in vector form. In principle, it would be possible to simply stack the atom coordinates into a vector and use that as the vector representation of the atomic structure. However, this is suboptimal. In general, a good representation should be invariant under physically invariant transformations of the atomic structure, such as translations, mirroring, and rotations. This can be achieved, for example, by constructing the vector representation based on the distances and angles between the atoms. One example of this approach are so called Coulomb matrices [38], where the values of the representation vector elements are determined by nuclear charges and interatomic distances. Another example is the many-body tensor representation (MBTR) that is explained in detail in Chapter 2.3.1.

Some machine learning methods do not require vector inputs, but rely on the one-to-one similarities between the inputs. One example of such a method is kernel ridge regression (KRR), which is also used in this work (see Chapter 2.3.2). These methods can also be used with atomic geometries. One option is to construct a function that compares any two geometries directly. Another, often simpler and more computationally efficient, option is to use a vector representation and define the similarity of two atomic geometries based on the distance between the vectors.

Representations of atomic structures can be global or local. Global representations are constructed based on the whole atomic system and they are used when

the target property is global, such as the energy of a molecule. Local descriptors are tied to a position in the system and can be used when predicting the value for a property at that point. They often consider only a small region of the whole system. Most commonly, local descriptors are used to represent the environment around an atom up to some distance. This is the case, for example, in many machine learning potentials where forces acting on an atom are predicted based on its environment [41, 42]. It should be noted, that even though local representations are more common for force prediction, global representations can be used as well. In fact, in this work forces are predicted with a model that uses MBTR, which is a global representation.

In step 3, a machine learning method is selected and trained to learn the mapping from atomic geometry to the property. Because a descriptor is used to represent the atomic structures in an accessible way, there are many powerful machine learning methods to choose from. For example, artificial neural networks have been used to predict potential energy surfaces [41]. In another study, many different machine learning methods, including neural networks and kernel ridge regression, were used to predict atomization energies of molecules [26].

The final step of creating a predictive machine learning model is quality control. Usually this is done by leaving part of the data set out of the model training and using it for testing. The model predictions are compared to the test data labels, giving an estimate for the prediction error of the model.

2.3 Machine Learning Model

The machine learning model in this work consists of two building blocks. The atomic structures are described in vector form using the many-body tensor representation (MBTR). After that, kernel ridge regression (KRR) is used for predicting the decomposition energies from the vectors. The main reason for choosing MBTR and

KRR was their good performance in previous similar works [26, 43, 44]. Additionally, these two methods are relatively simple, which makes differentiating the model easier. The selection process was also affected by the fact that good implementations for both methods are readily available. In this work, `DScribe` [45] implementation was used for MBTR and `scikit-learn` [46] for KRR.

2.3.1 Many-body Tensor Representation (MBTR)

MBTR [25] describes atomic geometries as distributions of small structural motifs that come in different atomic sizes k. The $k = 1$ term of MBTR considers single atoms and relates to the elemental content of the atomic structure. The $k = 2$ term considers atom pairs and their distances. The $k = 3$ term includes contributions from atom triplets and the angles between them. In the model that was used for this work, only the $k = 2$ term was used. The $k = 1$ term was deemed unnecessary due to the fact that the elemental contents of the structure are also indirectly included in the other terms. $k = 3$ term was left out in order to limit the complexity of the model and the number of hyperparameters. The partial feature space representation of an atomic structure is now given by

$$\mathrm{MBTR}^{Z_1,Z_2}(x) = \mathrm{MBTR}_2^{Z_1,Z_2}(x) = \sum_{l}^{|Z_1|} \sum_{m}^{|Z_2|} w^{l,m} d^{l,m}(x), \tag{2.5}$$

where Z_1 and Z_2 are two elements in the structure and the sums to $|Z_1|$ and $|Z_2|$ run over all the atom pairs that contain the two elements. $w^{l,m}$ is a weighting function that guarantees that the sum converges in periodic (and thus infinite) systems. It is defined as an exponential decay based on the distance of the atoms l and m :

$$w^{l,m} = e^{-s|\boldsymbol{R}_l - \boldsymbol{R}_m|}, \tag{2.6}$$

where $\boldsymbol{R}_l$ and $\boldsymbol{R}_m$ are the positions of the atoms and s is a positive scaling factor. The implementation of the weighting function gives rise to another parameter: the cutoff w_{cutoff}. When the distance between two atoms is so large that the value of

$w^{l,m}$ drops below w_{cutoff}, the pair no longer contributes to the sum and is ignored. The cutoff distance r_{cutoff} at which this happens can be determined through

$$w_{\text{cutoff}} = e^{-sr_{\text{cutoff}}} \Leftrightarrow r_{\text{cutoff}} = -\frac{\ln w_{\text{cutoff}}}{s}. \tag{2.7}$$

$d^{l,m}(x)$ is a Gaussian distribution function

$$d^{l,m}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-g^{l,m})^2}{2\sigma^2}}, \tag{2.8}$$

where g is a function that relates to the distances between the atoms. In this work, the inverse distance between the atoms was used:

$$g^{l,m} = \frac{1}{|\boldsymbol{R}_l - \boldsymbol{R}_m|}. \tag{2.9}$$

In 2.5, the MBTR representation is defined as a function of x. In order to use MBTR with KRR, distances between the MBTR representations of two different atomic structures are needed. An efficient way of doing this is to discretize $\text{MBTR}^{Z_1,Z_2}(x)$ by evaluating it only at a finite number of grid points. The grid is defined through three parameters: the number of grid points N_{grid}, lower limit x_{min}, and upper limit x_{max}.

$$\boldsymbol{x} = (x_{\text{min}}, x_{\text{min}} + \Delta x, \ \ldots, x_{\text{max}}), \tag{2.10}$$

where Δx is the grid spacing

$$\Delta x = \frac{x_{\text{max}} - x_{\text{min}}}{N_{\text{grid}} - 1}. \tag{2.11}$$

Now, the straightforward way of discretization would be to calculate the values of $d^{l,m}(x)$ at the grid points and sum them together as in 2.5. However, the norm of the distributions is preserved better with low grid densities, if the evaluations are done by differentiating the cumulative distributions of $d^{l,m}(x)$ numerically. The cumulative distribution function of the Gaussian distribution is

$$D^{l,m}(x) = \int_{-\infty}^{x} d^{l,m}(x')dx' = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x - g^{l,m}}{\sigma\sqrt{2}}\right)\right], \tag{2.12}$$

where erf is the error function. $d^{l,m}(x)$ can then be estimated by evaluating $D^{l,m}(x)$ at grid points that are offset from $\boldsymbol{x}$ by $\Delta x/2$:

$$d^{l,m}(x) = \frac{D^{l,m}(x + \frac{\Delta x}{2}) - D^{l,m}(x - \frac{\Delta x}{2})}{\Delta x}. \tag{2.13}$$

In discretized form, $d^{l,m}(x)$ can be written as a vector $\boldsymbol{d}^{l,m}$: $\boldsymbol{d}_i^{l,m} = d^{l,m}(\boldsymbol{x_i})$. The discretized partial MBTR is now given by

$$\boldsymbol{M}^{Z_1,Z_2} = \sum_{l}^{|Z_1|} \sum_{m}^{|Z_2|} w^{l,m} \boldsymbol{d}^{l,m}. \tag{2.14}$$

The full feature space vector representation $\boldsymbol{M}(\boldsymbol{s})$ of an atomic structure $\boldsymbol{s}$ is obtained by stacking the contributions from all possible element pairs Z_1 and Z_2 after each other. Figure 2.1 visualizes an example MBTR representation of a $\mathrm{CsPb(Cl, Br)_3}$ structure.

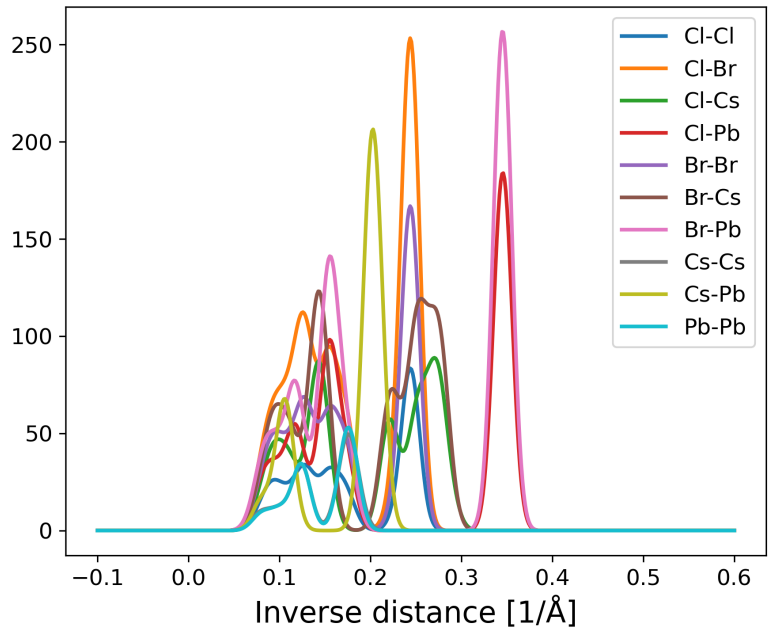


Figure 2.1: Example MBTR representation of a $\mathrm{CsPb(Cl, Br)_3}$ structure with hyperparameters $x_{\mathrm{min}} = -0.1$, $x_{\mathrm{max}} = 0.6$, $N_{\mathrm{grid}} = 500$, $\sigma = 0.01$, $s = 0.5$, and $w_{\mathrm{cutoff}} = 0.001$.

2.3.2 Kernel Ridge Regression (KRR)

As its name suggests, KRR is based on ridge regression [47], that is, linear regression with L2-norm regularization. KRR is obtained by using so called kernel trick to introduce nonlinearity to the model. The decomposition energy (ΔH) prediction of the model for an atomic structure $\boldsymbol{s}$ is a weighted sum of kernel functions

$$\Delta H^{\text{pred}}(\boldsymbol{s}) = \sum_{i}^{N} \beta_i k(\boldsymbol{s}, \boldsymbol{s}_i), \tag{2.15}$$

where $\boldsymbol{s}_i$ are a set of reference structures and β_i are regression coefficients. k is a kernel function that measures the similarity of two atomic structures. In this work, a Gaussian kernel function was used with the MBTR representations.

$$k(\boldsymbol{s}, \boldsymbol{s}') = e^{-\gamma ||\boldsymbol{M}(\boldsymbol{s}) - \boldsymbol{M}(\boldsymbol{s}')||_2^2} \tag{2.16}$$

The regression coefficients can be fitted to a reference data set with known decomposition energies ΔH_i^{ref}. The loss function of the KRR model is

$$L = \sum_{j}^{N} \left(\Delta H^{\text{pred}}(\boldsymbol{s}_j) - \Delta H_j^{\text{ref}} \right)^2 + \alpha \boldsymbol{\beta}^T \boldsymbol{K} \boldsymbol{\beta}, \tag{2.17}$$

where $\boldsymbol{K}$ is the kernel matrix $K_{i,j} := k(\boldsymbol{s}_i, \boldsymbol{s}_j)$ and α is a regularization parameter. The loss can be minimized in terms of $\boldsymbol{\beta}$, which gives the fitted coefficients

$$\boldsymbol{\beta} = (\boldsymbol{K} + \alpha \boldsymbol{I})^{-1} \boldsymbol{\Delta H}^{\text{ref}}. \tag{2.18}$$

3. Data

Machine learning is a data driven approach. If the data that is used to train and test the model is not of high quality, the results cannot be either. Most importantly, in a physical problem the data has to reflect reality. Secondly, the data needs to be representative of the problem in question.

The data that was used in this work builds on the previous work of Jingrui Li and Patrick Rinke [48, 24]. In this chapter, I introduce two data sets that were generated by Li. Both data sets consist of 40 atom $CsPb(Cl, Br)_3$ perovskite structures. The structures in **Dataset 1** were generated algorithmically. The data is used for training the machine learning model, which is why the diversity of the data set was of high priority in the generation. The stability of the generated structures was carefully evaluated with DFT calculations. **Dataset 2** consists of 100 DFT-driven structural optimization trajectories starting from geometries that were randomly sampled from **Dataset 1**. In Chapter 5, the machine learning model is used for structural optimizations and the results are compared to **Dataset 2**.

3.1 Dataset 1

3.1.1 Atomic Structures

The first data set consists of algorithmically generated $CsPb(Cl, Br)_3$ perovskite geometries. Its purpose is to be used as training data for the machine learning model,

before the model is used for structural optimization. The training data should be representative of all kinds of atomic structures that the model can encounter during the structural optimizations. This is why, it is important that the training data is diverse in terms of deformations and elemental substitutions.

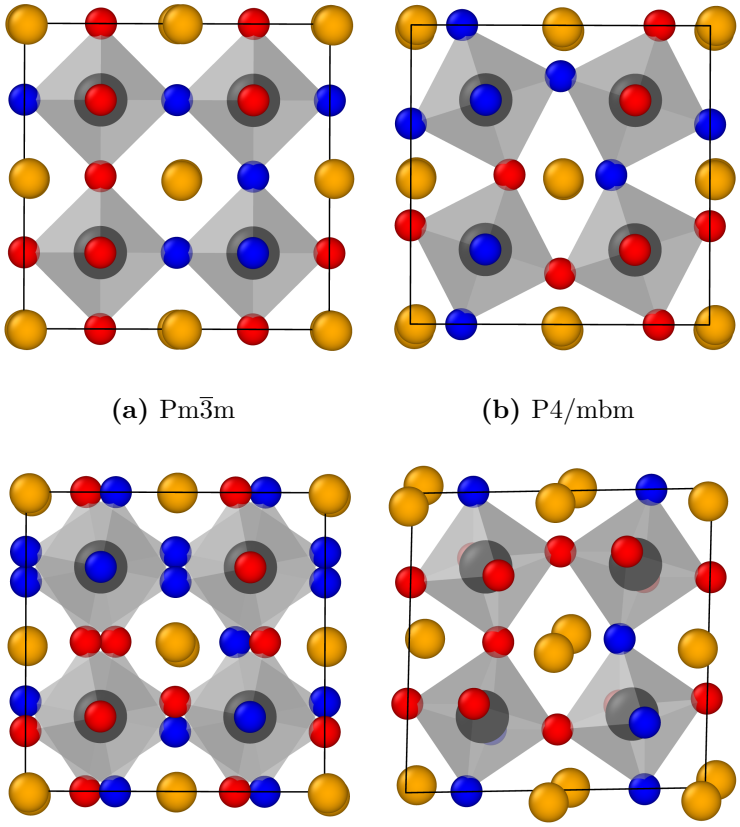
The first way of increasing the diversity was to include structures of four different perovskite phases: $Pm\bar{3}m$, P4/mbm, I4/mcm, and Pnma. Even more deformations were introduced to the data by randomizing the amount of octahedral tilting as well as the Cs positions. The need for different elemental substitution levels was met by randomizing the X-site anion elements in each generated structure.

The generation process of the structures had the following steps:

1. Pick Cl concentration between 0 and 1 randomly and assign the elements of corresponding numbers of X-site atoms to Cl and Br
2. Permute the X-site atoms randomly
3. Set the element of each X-site anion randomly to Cl or Br
4. Interpolate the lattice dimensions and atom positions based on relaxed pure $CsPbCl_3$ and $CsPbBr_3$ according to Vegard's law
5. Introduce small random deviations to Cs positions
6. Deviate the octahedral rotation angles randomly

The fifth step was not performed for the $Pm\bar{3}m$ structures that have no octahedral tilting. We generated 1000 structures for each of the four phases, ending up with 4000 structures in total. Examples of each phase are shown in Figure 3.1. The structures are $2 \times 2 \times 2$ perovskite supercells with 40 atoms in each geometry.

I analyzed the results of the generation process by plotting the Cl concentrations of all generated structures (see Figure 3.2). The concentration follows approximately binomial distribution, peaking at 0.5 and tapering off closer to 0.0 and



(a) Pm$\bar{3}$m **(b)** P4/mbm

(c) I4/mcm **(d)** Pnma

Figure 3.1: Examples of atomic structures in **Dataset 1**. The structures are shown from the $c-$direction.

1.0. Most of the substitution range is covered, but there are no structures with concentrations beyond 0.15 and 0.85.

Next, I analyzed the octahedral rotation angles in the generated structures. The rotation angles were estimated by comparing the directions of the octahedron diagonals to the lattice vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$. The results of the analysis are shown in Figure 3.3. Pm$\bar{3}$m is not included because it has no octahedral tilting in any direction. The same goes for R_a and R_b of P4/mbm and I4/mcm. The R_b rotation of Pnma structures was also omitted because it is identical to R_a.

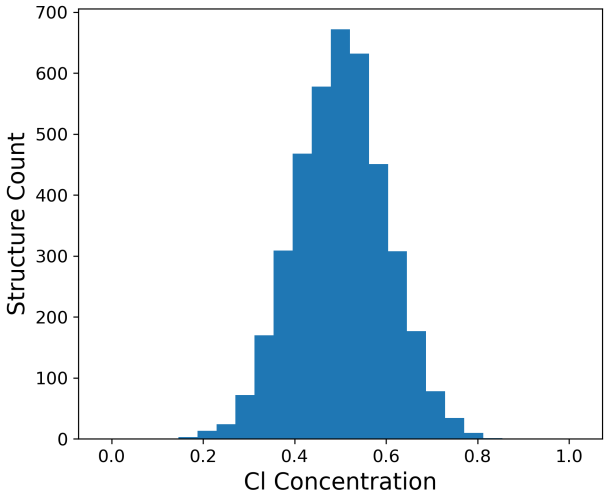
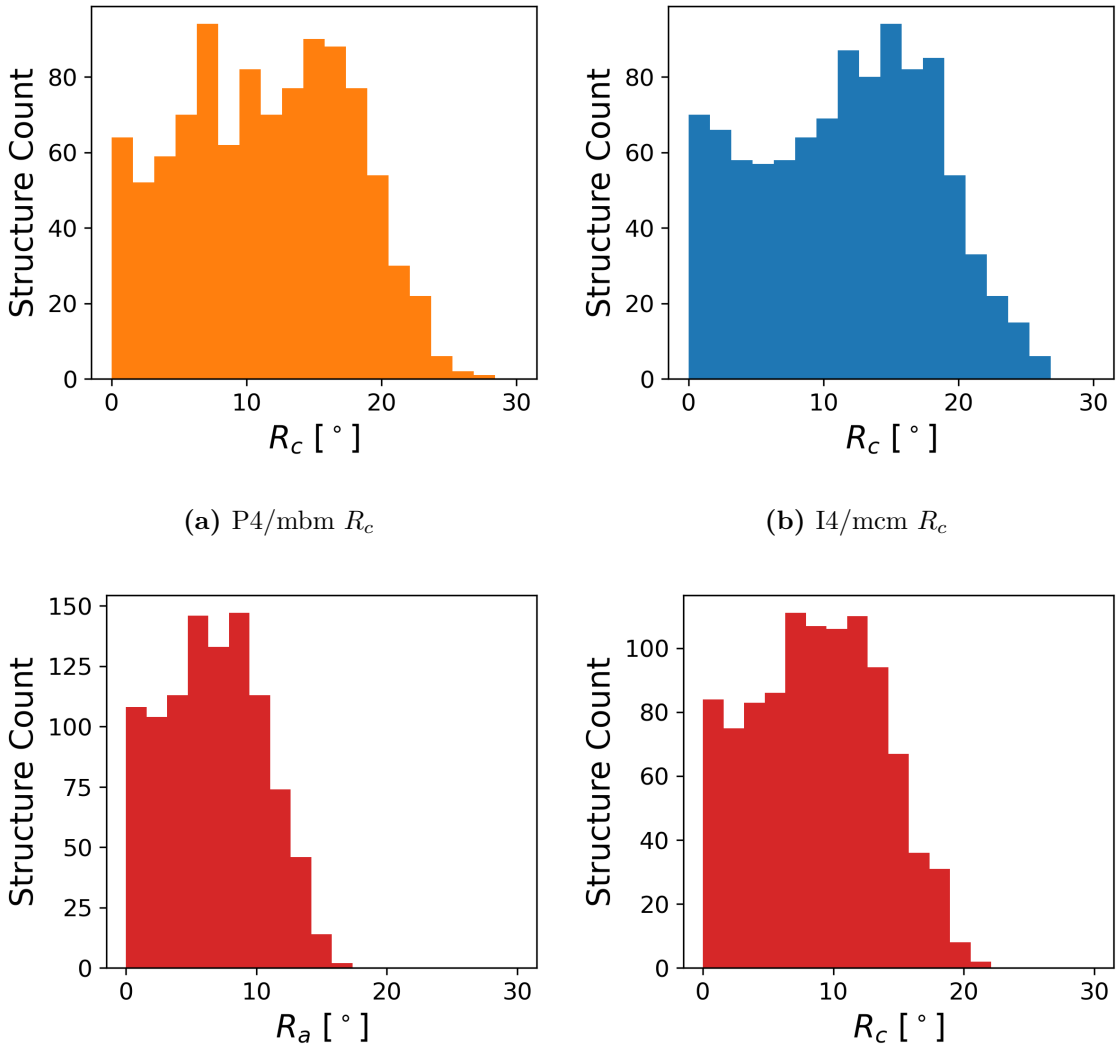


Figure 3.2: Cl concentrations in **Dataset 1**.

3.1.2 DFT Calculations

DFT was used to calculate the total energies $E^{\mathrm{DFT}}(\boldsymbol{s})$ of all the structures in **Dataset 1**. The DFT code that we used was `FHI-aims` [35]. Here is a list of the key settings that were used in the calculations:

- Exchange-correlation functional: PBEsol [49]
- Relativistic treatments: "Atomic ZORA"
- Self-consistency accuracy of E_{tot}: 1×10^{-6} eV
- Integration grid: "Tight"
- k-grid: $4 \times 4 \times 4$
- Basis set:
  - Br: "First tier"
  - Cl: "First tier" + hydro 3d
  - Cs: "First tier" + hydro 4d
  - Pb: "First tier"



(a) P4/mbm R_c

(b) I4/mcm R_c

(c) Pnma R_a

(d) Pnma R_c

Figure 3.3: Octahedral rotation in **Dataset 1**.

3.1.3 Decomposition Energy

The decomposition energy ΔH is the amount of energy that is required to break the perovskite structures into its constituents. Thus, it is a measure of stability. A negative ΔH indicates that the material will not decompose. Materials with lower ΔH values are more stable. This is why the decomposition energy is a good label to use in the training of the machine learning model. If the model can learn the

relationship between the atomic geometry and ΔH, it can be used to predict the stability of materials.

In this work ΔH was defined as

$$\begin{aligned}\Delta H =& E[\mathrm{CsPb(Cl}_x\mathrm{Br}_{3-x})] - aE[\mathrm{CsCl}] - (1-a)E[\mathrm{CsBr}] \\ & - \frac{x-a}{2}E[\mathrm{PbCl_2}] - \left(1 - \frac{x-a}{2}\right)E[\mathrm{PbBr_2}],\end{aligned} \tag{3.1}$$

where CsCl, CsBr, $PbCl_2$, and $PbBr_2$ are the possible decomposition products of $\mathrm{CsPb(Cl}_x\mathrm{Br}_{3-x})$ and E the corresponding total energies. $E[\mathrm{CsPb(Cl}_x\mathrm{Br}_{3-x})]$ is the total energy of the composite material per perovskite unit, that is per five atoms. a controls the fractions of the decomposition products and is bounded by

$$0 \leq a \leq x \quad \text{and} \quad 0 \leq (1-a) \leq 3-x. \tag{3.2}$$

The decomposition energies of the generated atomic structures were found using the total energy DFT calculation results from the previous section. The energies of the decomposition products were calculated with DFT as well. The decomposition energy of an atomic structure is

$$\begin{aligned}\Delta H^{\mathrm{DFT}}(\boldsymbol{s}) =& \frac{1}{8}E^{\mathrm{DFT}}(\boldsymbol{s}) - a_{\max}E^{\mathrm{DFT}}(\mathrm{CsCl}) - (1-a_{\max})E^{\mathrm{DFT}}(\mathrm{CsBr}) \\ & - \frac{x-a_{\max}}{2}E^{\mathrm{DFT}}(\mathrm{PbCl_2}) - \left(1 - \frac{x-a_{\max}}{2}\right)E^{\mathrm{DFT}}(\mathrm{PbBr_2}),\end{aligned} \tag{3.3}$$

where the total DFT energy of the structure has been divided by eight to get the energy per one perovskite unit. $a_{\max}$ is the value of a that maximizes ΔH within the limitations (3.2). This has to be done to guarantee that the decomposition process that requires the least energy is considered. Because

$$\frac{\partial \Delta H}{\partial a} = - E^{\mathrm{DFT}}(\mathrm{CsCl}) + E^{\mathrm{DFT}}(\mathrm{CsBr}) + \frac{1}{2}E^{\mathrm{DFT}}(\mathrm{PbCl_2}) - \frac{1}{2}E^{\mathrm{DFT}}(\mathrm{PbBr_2}) \tag{3.4}$$

$$= -0.024\,\mathrm{eV} < 0, \tag{3.5}$$

a that maximizes ΔH within the limits is always

$$a_{\max} = \max(0, x-2). \tag{3.6}$$

It is noteworthy that in (3.3) the only term of $\Delta H^{\mathrm{DFT}}(\boldsymbol{s})$ that depends on the structure is $\frac{1}{8}E^{\mathrm{DFT}}(\boldsymbol{s})$. This means that the total energy gradients can be written in terms of decomposition energy gradients:

$$\nabla E^{\mathrm{DFT}}(\boldsymbol{s}) = 8\nabla\left(\Delta H^{\mathrm{DFT}}(\boldsymbol{s})\right). \tag{3.7}$$

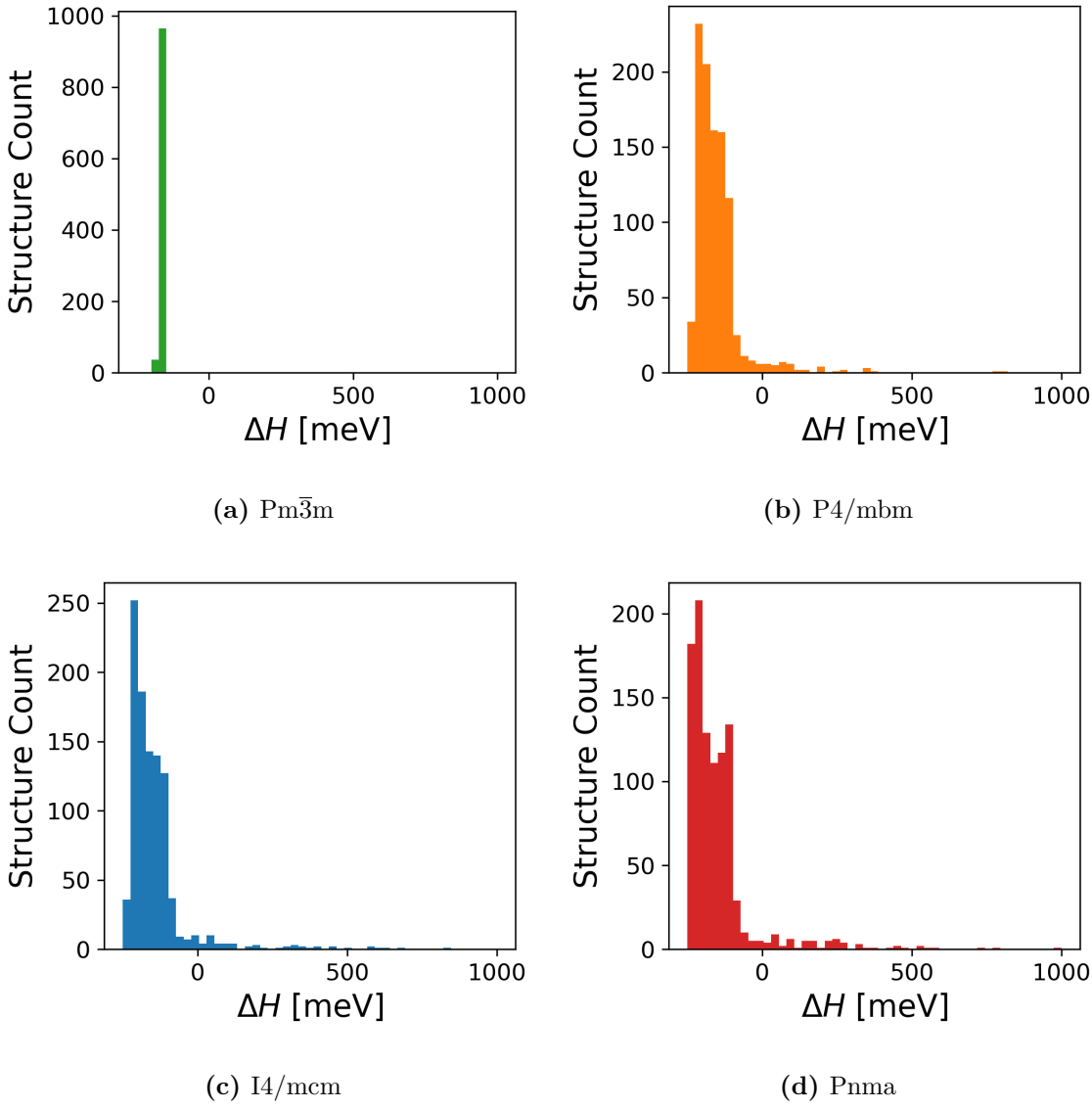
This allows using the ΔH predicting model in structural optimizations.

ΔH were calculated for all the structures in **Dataset 1**. The energy distributions can be seen in Figure 3.4. The energies of the $Pm\bar{3}m$ structures are highly peaked between $-185\,\mathrm{meV}$ and $-155\,\mathrm{meV}$. The energy ranges of the other three phases are larger, with most energies between $-250\,\mathrm{meV}$ and $0\,\mathrm{meV}$, and few unstable outliers.

3.2 Dataset 2

The second data set consists of structural optimization trajectories. Its purpose is to be a reference to which the machine learning model-driven optimizations can be compared. We took 100 structures from **Dataset 1** as starting points of the optimization runs. 50 structures were picked from both $Pm\bar{3}m$ and Pnma randomly. The optimizations were performed in `FHI-aims` as local optimizations of total energy using the BFGS algorithm. The settings for the DFT calculations were the same as with **Dataset 1**. The lattice vectors were fixed during the optimizations, meaning that only the atom positions were allowed to change. The mean force convergence limit for the optimizations was $5 \times 10^{-3}\,\mathrm{eV/Å}$.

In total, 3604 atomic geometries were reached during the 100 optimization runs. Decomposition energies were calculated for all the structures the same way as with **Dataset 1**. The change in the decomposition values as the optimizations progress are shown in Figure 3.5. The mean decomposition energy drops from $-162\,\mathrm{meV}$ to $-280\,\mathrm{meV}$ with the $Pm\bar{3}m$ structures. For Pnma, the energy decrease



(a) Pm$\bar{3}$m **(b)** P4/mbm

(c) I4/mcm **(d)** Pnma

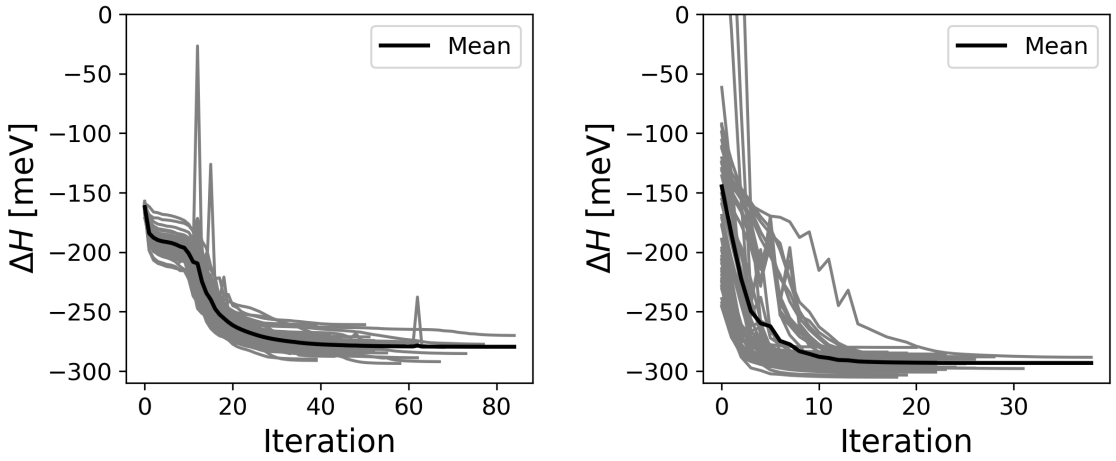
Figure 3.4: Decomposition energies in **Dataset 1**.

is from −145 meV to −293 meV. It is noteworthy that the final energies are largely outside the energy range of the structures in **Dataset 1**.

On top of energies, also the DFT calculated atomic forces are known for all the structures in **Dataset 2**. This helps greatly because they can be used a reference when the the machine learning models energy gradient prediction is tested.

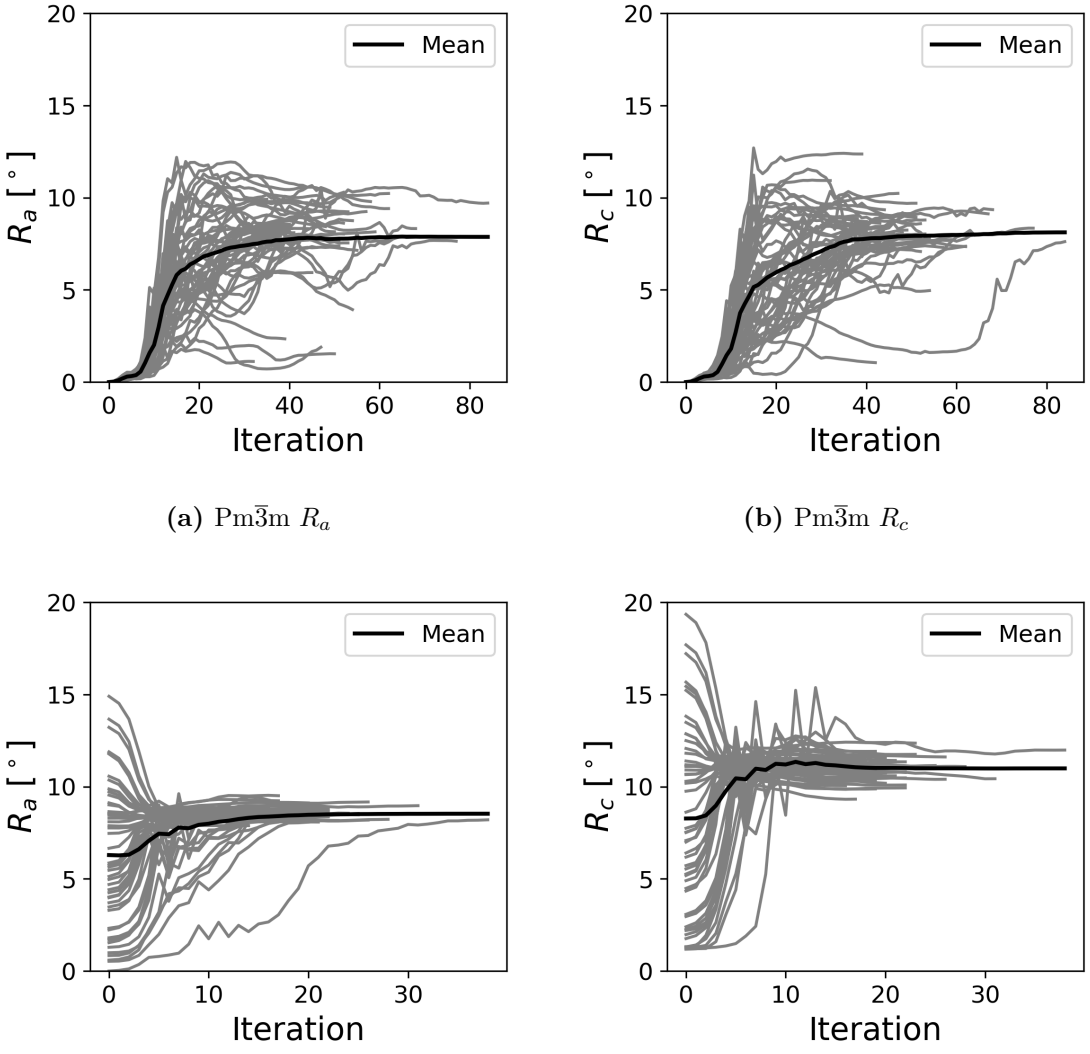
Next, I analyzed how the atomic structures change during the optimizations. The octahedral rotation angles were calculated for all the structures that were

reached. The results are shown in Figure 3.6. The first observation is that the $Pm\bar{3}m$ structures start to show octahedral tilting in all directions when the optimizations proceed. Another observation is that the rotational angles Pnma structures converge. Initially, the range of the rotation angles is spreading from 0° to 20°, whereas the final rotation angles are all just a couple of degrees apart.



(a) $Pm\bar{3}m$ **(b)** Pnma

Figure 3.5: Decomposition energy in structural optimization trajectories of **Dataset 2**.



(a) Pm$\bar{3}$m R_a

(b) Pm$\bar{3}$m R_c

(c) Pnma R_a

(d) Pnma R_c

Figure 3.6: Octahedral rotation in structural optimization trajectories of **Dataset 2**.

4. Model Selection

The machine learning model that was defined in Chapter 2.3 can be used for predicting the decomposition energies of atomic structures. Varying the hyperparameter values of the model creates new models with different prediction accuracies. Model selection is a process of sampling these models and picking the ones that perform the best.

The main steps of model selection are shown in Figure 4.1. **Dataset 1** was used for fitting the models and testing their performance. Out of its 4000 atomic structures, 3200 were placed into the training set, while the remaining 800 form the test set. The data was split randomly in such a way that the different phases are equally represented in both subsets. The training data structures were used for fitting and validating the performance of the different models that were sampled during the hyperparameter optimization. In this chapter, a tailored Bayesian optimization-based method is presented as means to efficiently find the optimal hyperparameter values. The test set structures were used to perform the final performance evaluation on the selected models in an unbiased way.

4.1 Hyperparameter Optimization

The machine learning model that was introduced in Chapter 2.3 has eight hyperparameters in total: six MBTR parameters and two KRR parameters. They control the training process of the model, which means that unlike other parameters they

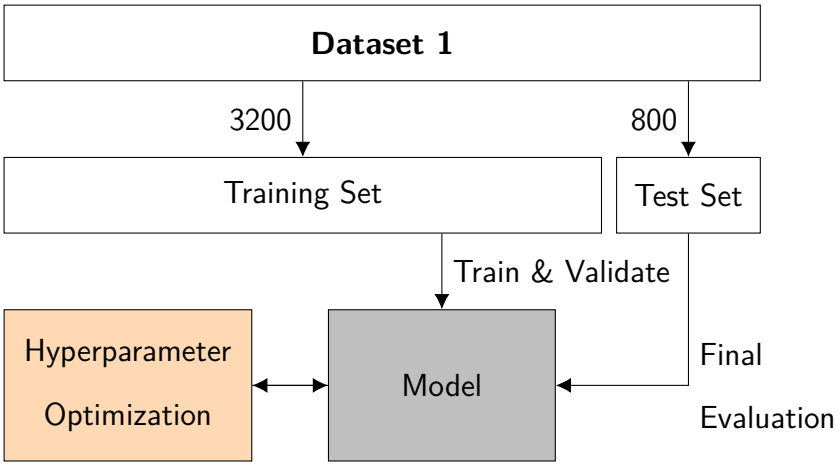


Figure 4.1: Model selection steps. **Dataset 1** is divided into training and test sets. The hyperparameters are optimized by training the model repeatedly on the training data with different hyperparameter values. The test data is used after the optimization to evaluate the accuracy of the final model.

cannot be optimized directly via training. Instead, the optimal hyperparameters can be found by training the model repeatedly with different hyperparameter values and comparing the performance of the resulting models. In this section, I list all the hyperparameters and how they were treated in the optimization. Some of the parameters can be fixed in advance. Then, the method used for optimizing the remaining parameters and the results from applying it are presented.

4.1.1 Hyperparameters

The six MBTR hyperparameters are x_{min}, x_{max}, N_{grid}, w_{cutoff}, s, and σ. Parameters x_{min} and x_{max} control the span of the MBTR discretization grid. Analyzing the data reveals that the smallest distances between atoms in the structures are between bonded Pb and X site anions. This distance is about 2.87 Å, which corresponds to an inverse distance of $0.35\,\text{Å}^{-1}$. Also, the inverse distances can never be negative. The limits of the grid should be large enough to leave space for the Gaussian smoothing around the exact values. Taking all considerations into account, the parameters

were fixed to $x_{\text{min}} = -0.1\,\text{Å}^{-1}$ and $x_{\text{max}} = 0.6\,\text{Å}^{-1}$.

The MBTR vector representations are used in the machine learning model for calculating the distances between the structures (see equation (2.16)). The MBTR parameter N_{grid} only affects the distance estimation accuracy by controlling how smoothly the discretized representation follows the underlying function. Optimizing N_{grid} purely in terms of predictive accuracy is not meaningful because increasing it should only ever improve the accuracy of the model. The solution was to fix N_{grid} to a large but computationally manageable value of 1000.

Like N_{grid}, w_{cutoff} also affects the estimation accuracy of the distances between the atomic structures. It is the weight threshold below which the contribution from a pair of atoms is not included in the representation. A low w_{cutoff} makes the transition from the contributing to the noncontributing part of the inter-atomic space smoother, and thus increases the accuracy of the distance estimation. However, increasing w_{cutoff} makes the MBTR computation more efficient. We are faced with a classic trade-off between accuracy and efficiency. The solution, after testing, is to fix w_{cutoff} to a sufficiently low value of $w_{\text{cutoff}} = 10^{-3}$.

According to equation (2.7), after fixing w_{cutoff}, the cutoff radius of contributing atom pairs is only affected by the parameter s. Also for s I was faced with trade-off considerations. However, in this case I did not fix s to a single value. Instead, I tested three different values to create three different tiers of MBTR representations with their own accuracy and efficiency. The three $s-$values were selected based on their corresponding cutoff radii r_{cutoff}. The first and most efficient tier, **Tier 1**, has a cutoff radius that spans little more than one perovskite unit cell $r_{\text{cutoff}} = 6.27\,\text{Å}$. This value was obtained by taking the mean lattice vector length in the structures of **Dataset 1** and adding 10% to it. The least efficient tier, **Tier 3**, has a cutoff radius that is twice as large $r_{\text{cutoff}} = 12.54\,\text{Å}$. **Tier 2** has a cutoff radius that is exactly in the middle of the other two $r_{\text{cutoff}} = 9.41\,\text{Å}$. Details about the tiers have

	r_{cutoff} [Å]	s	t_{MBTR} [ms]
Tier 1	6.27	1.10	36
Tier 2	9.41	0.73	120
Tier 3	12.54	0.55	210

Table 4.1: The three hyperparameter tiers and the corresponding cutoff radii r_{cutoff} and weighting parameter s values. The final column of the table shows the average MBTR vector computation times per one atomic structure.

been compiled in Table 4.1. The table also has the MBTR vector computation times for the tiers. The computation time for **Tier 3** is almost six times longer than for **Tier 1**, which is a significant difference considering the fact that the feature vector calculation is the most time consuming part of the model.

The only MBTR parameter that is left unfixed is the Gaussian smoothing parameter σ. Unlike other MBTR parameters, it is expected to have a well defined minimum. Too high a value will result in loss off information due to overlapping Gaussian peaks, whereas too low a value will result in overly ragged and peaked MBTRs. The optimal value is found somewhere between the two extremes. Likewise, KRR parameters α and γ cannot be fixed in advance. This leaves three hyperparameters that need to be optimized: σ, α, and γ.

4.1.2 Optimization Method

To select between the different models in hyperparameter optimization, I defined a function that quantifies the training quality. Here, the mean absolute error (MAE) of decomposition energy predictions was used:

$$\text{MAE}[\Delta H] = \frac{1}{N}\sum_{i}^{N} |\Delta H_i^{\text{pred}} - \Delta H_i^{\text{DFT}}|, \tag{4.1}$$

where ΔH_i^{pred} is the model prediction for the decomposition energy of the ith structure in a set of N structures and ΔH_i^{DFT} is the corresponding DFT energy. The

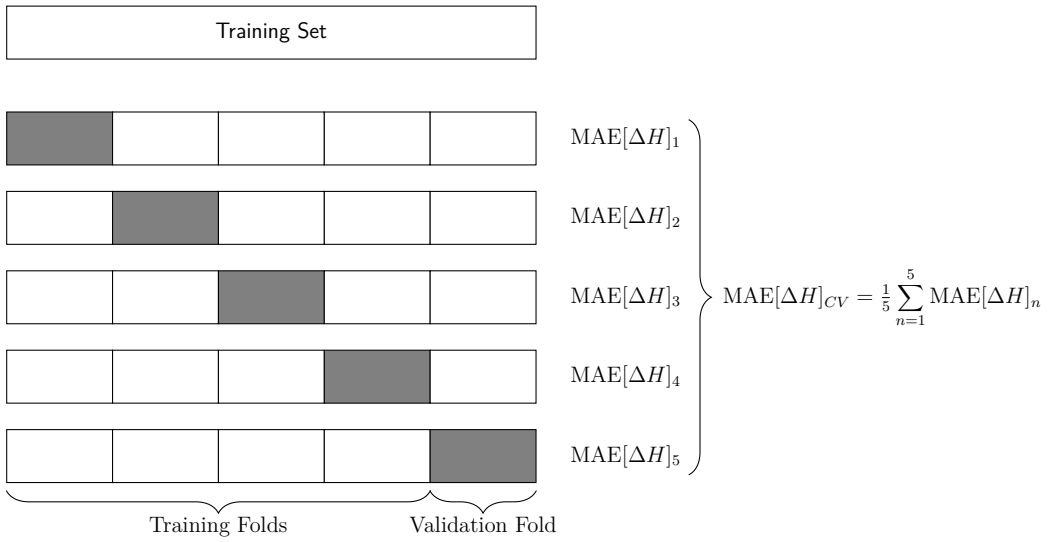


Figure 4.2: Calculation of the validation error $\mathrm{MAE}[\Delta H]_{CV}$ using 5-fold cross validation

simplest way of estimating the performance would be to split the training set into two parts and use one of them to train the model and the other one to validate the fit by calculating $\mathrm{MAE}[\Delta H]$. However, this approach suffers from the inevitable variability rising from the choice of the data split. Using cross-validation mitigates this problem by fitting the model multiple times. In this work, 5-fold cross-validation was used. Its working principle is shown in Figure 4.2. The training set was divided into five equal folds randomly in such a way that all four phases are equally represented in the folds. The model is then fitted five times. On each fit, four folds are used for training and one fold for calculating the validation error $\mathrm{MAE}[\Delta H]_n$. The validation fold is different for each fit. The final validation error is found by averaging over the the fits:

$$\mathrm{MAE}[\Delta H]_{CV} = \frac{1}{5}\sum_{n=1}^{5}\mathrm{MAE}[\Delta H]_n. \tag{4.2}$$

This function can now be used to map σ, α, and γ values on the prediction error of the model. The hyperparameter optimization problem becomes simply a three-dimensional function minimization problem of $\mathrm{MAE}[\Delta H]_{CV}$.

It would be possible to naively minimize $\mathrm{MAE}[\Delta H]_{CV}$ in all three dimensions

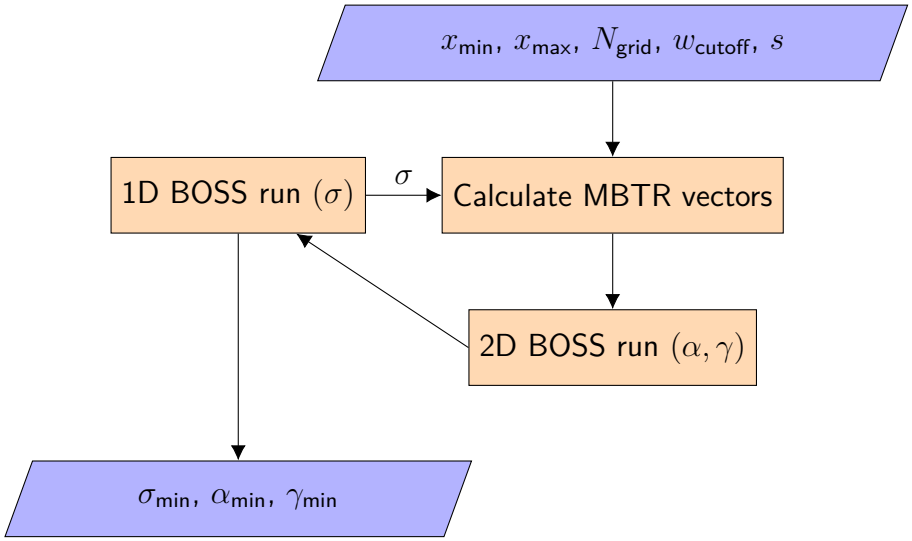


Figure 4.3: σ-optimization

simultaneously but a more efficient approach can be taken. Calculating MBTR vectors is the most time consuming part of evaluating $\text{MAE}[\Delta H]_{CV}$, whereas fitting KRR is fast. For this reason, I adopted an approach in which the optimal KRR parameter values are found for each value of σ. The result is a two-layered optimization sequence in which an iteration of $\sigma-$optimization is followed by MBTR vector calculation and multiple iterations of optimizing α and γ. Because the optimization converges with fewer iterations in one dimension than it would in three, fewer MBTR vector recalculations are needed, making the method more efficient.

The optimization process is visualized in Figure 4.3. Both layers of the optimization were done using the Bayesian optimization code BOSS [50, 51]. The main component is a one-dimensional BOSS optimization for σ. The search was limited to $\log_{10}\sigma \in [-3, 0]$. 20 values were sampled in total: 5 initial points and 15 acquisitions with eLCB function. Each $\sigma-$value that BOSS sampled was passed to the MBTR vector calculation, after which the optimal KRR parameters α_{min} and γ_{min} that correspond to the sampled σ were found with a separate BOSS run. These 2D BOSS optimizations were ran for 40 error evaluations: 10 initial points and 30 acquisitions with the eLCB function. The adequacy of acquisitions was tested by

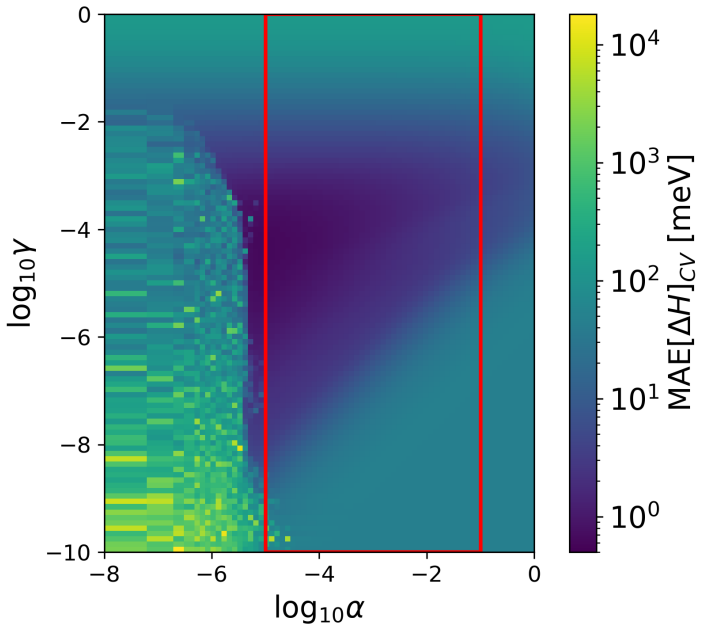


Figure 4.4: MAE$[\Delta H]_{CV}$ surface in $(\alpha, \gamma)-$space with $\sigma = 5.0 \times 10^{-2}$. The updated search limits for the parameters are visualized with the red rectangle.

running test optimizations with different σ values and checking that all the runs had converged well within the limit. The optimal KRR parameter values were passed back to the main BOSS run where another iteration step can now be taken. The outputs of the whole process are σ_{min}, α_{min}, and γ_{min} : the optimal values for all unfixed hyperparameters. The whole optimization process needed to be repeated three times to find the optimal hyperparameters for the three different cutoff tiers.

4.1.3 Optimization Results

The hyperparameter optimization steps were repeated for the three tiers. During the optimization, many $(\alpha, \gamma)-$optimizations were run. Figure 4.4 shows a typical MAE surface in $(\alpha, \gamma)-$space. The error has a broad minimum around $(\log_{10}\alpha, \log_{10}\gamma) = (-4, -5)$. The surface is well behaved for $\log_{10}\alpha \gtrsim -5.3$, below which the error values rise sharply. As can be seen from equation (2.18), fitting the KRR model requires the inversion of the $(\boldsymbol{K} + \alpha\boldsymbol{I})$-matrix. The matrix becomes

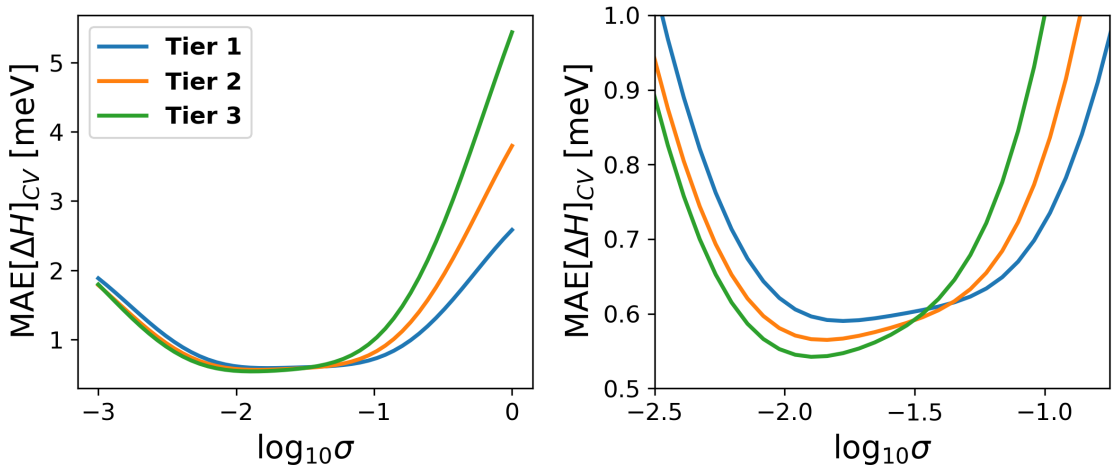


Figure 4.5: (a) shows the validation error $\mathrm{MAE}[\Delta H]_{CV}$ as a function of $\log_{10}\sigma$ for the three tiers. (b) is the same plot cropped in a way that shows the minima better.

ill-conditioned for low α, which makes the inversion more difficult and triggers the use of an alternative fitting method in the KRR implementation. With the change of the method, the fit quality drops. The sharp edge on the error surface also led to convergence problems in the (α, γ) BOSS runs.

The problem was avoided by restricting the parameter search to $\log_{10}\alpha \in [-5, -1]$ and $\log_{10}\gamma \in [-10, 0]$, leaving out the area in which the alternative fitting method was triggered. With this update of the optimization method, convergence was not a problem anymore. α_{min} was in most cases found to be at $\log_{10}\alpha = -5$ but this is a small compromise due to the flatness of the validation error around the optimum. The optimum never reached any of the other limits, indicating that they were sufficient.

The optimization results are shown in Figure 4.5. All the tiers perform similarly. The validation error $\mathrm{MAE}[\Delta H]_{CV}$ is very flat around the minimum, reaching values below 1 meV all the way from $\log_{10}\sigma \approx -2.5$ to $\log_{10}\sigma \approx -1.0$. The found σ_{min} decreases a little with the higher tiers.

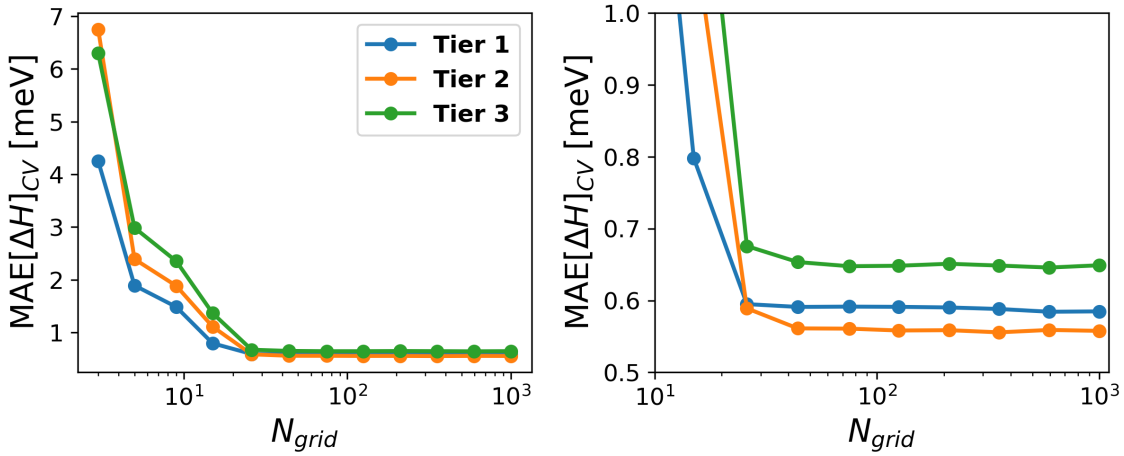


Figure 4.6: (a) shows the validation error MAE$[\Delta H]_{CV}$ as a function of N_{grid} for the three tiers. (b) is the same plot cropped in a way that shows the convergence better.

Next, in order to improve the efficiency of the models, the possibility of using lower N_{grid} was investigated. This was done, simply, by fixing σ to the optimum values that were found in the σ-optimization, and calculating MAE$[\Delta H]_{CV}$ with different N_{grid} values. 12 N_{grid} values were sampled with geometric spacing in the range $[3, 1000]$. For each value the KRR parameters α and γ were reoptimized using the same 2D BOSS setup as before. This is crucial because especially γ_{min} is highly dependent on the amount of grid points in the MBTR representation.

The results of the N_{grid} check are shown in Figure 4.6 and they look mostly as expected. On the lowest N_{grid} values, MAE$[\Delta H]_{CV}$ values are very high. When the values increase, however, the error converges. There is no significant improvement in the model performance in any tier after $N_{\text{grid}} \approx 30$. The choice for the final N_{grid} values was done conservatively and they were set to $N_{\text{grid}} = 50$ for all tiers. To conclude the hyperparameter optimization, (α, γ)-optimization was performed one more time for each tier. The final hyperparameters have been compiled in Table 4.2 along with the validation error values that were reached using them.

	σ_{min}	N_{grid}	α_{min}	γ_{min}	MAE$[\Delta H]_{CV}$ [meV]
Tier 1	17.7e-2	50	1.0e-5	1.53e-4	0.59
Tier 2	14.2e-2	50	1.0e-5	7.34e-6	0.56
Tier 3	13.3e-2	50	1.0e-5	1.16e-6	0.53

Table 4.2: The optimal hyperparameter values for different model tiers. The last column has validation errors calculated with the optimal parameters.

4.2 Decomposition Energy Predictions

Now, the three differently tiered models that were selected are almost ready for decomposition energy prediction. However, up until this point the models have only been fitted on four fifths of the available training data because in cross-validation one fifth was always left out for validating the prediction accuracy. Now, that is not necessary anymore and the models were slightly improved by fitting them on all 3200 structures of the training set.

The accuracy of these newly fitted models was evaluated by predicting the decomposition energies ΔH^{pred} of the 800 test set structures and comparing the predicted values to the DFT energies ΔH^{DFT}. The mean absolute errors have been compiled in Table 4.3, where the errors are given separately for the four phases. There are three main observations to make. Firstly, the performance of all model tiers is very similar. **Tier 3** is slightly better than **Tier 1** for Pm$\bar{3}$m, I4/mcm, and P4/mbm, whereas the opposite is true for Pnma. Secondly, the results are very impressive overall. Here the decomposition energies and thus the errors are given per one perovskite unit, that is five atoms. This means that the error per atom is about 0.1 meV.

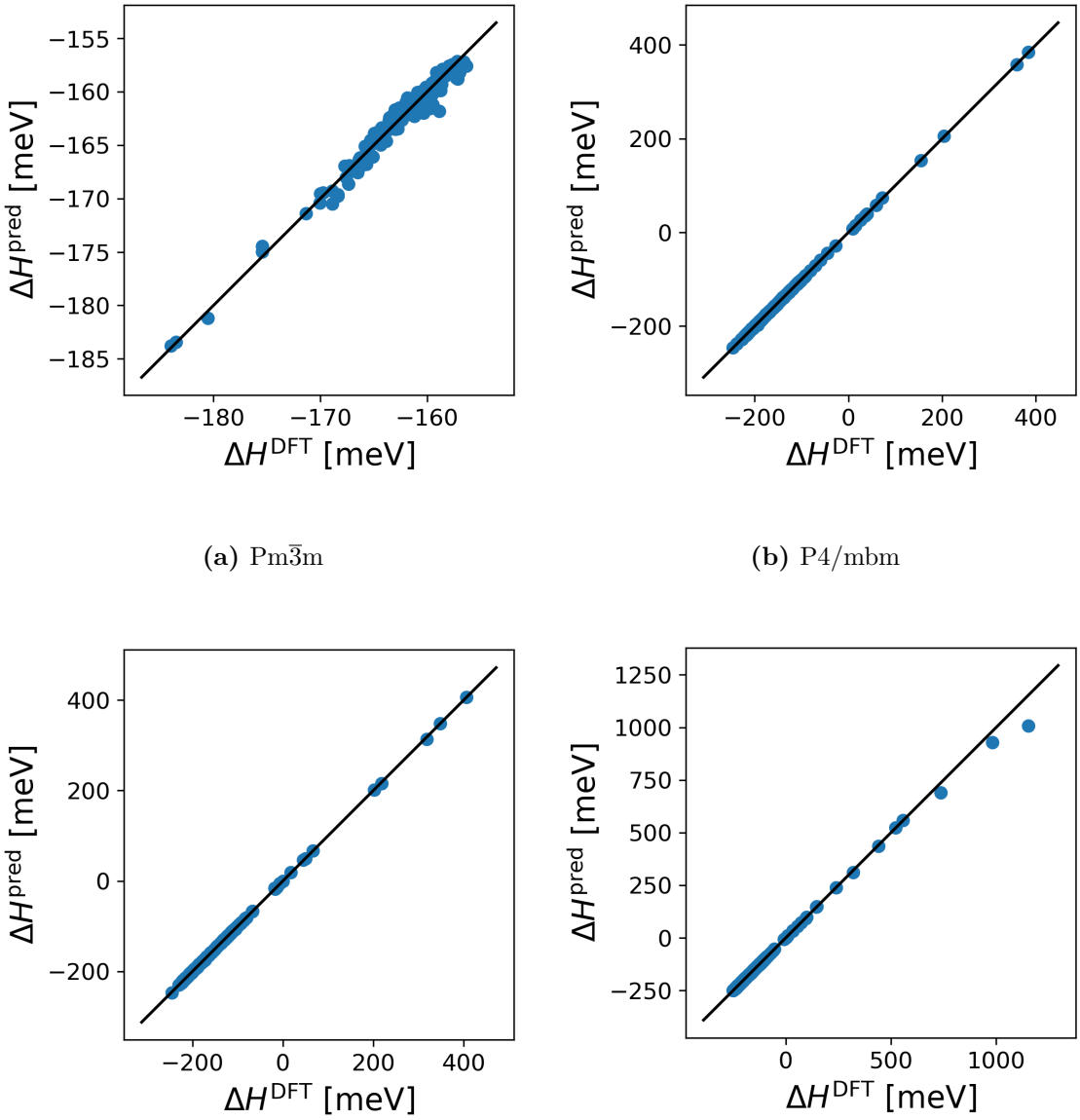
The third observation is that the error for the Pnma structures is multiple times larger than what it is for the other phases. The reason for this becomes obvious when looking at the ΔH^{DFT} vs. ΔH^{pred} plots in Figure 4.7. Like with other phases,

	$Pm\bar{3}m$	I4/mcm	P4/mbm	Pnma
Tier 1	0.54	0.54	0.48	1.79
Tier 2	0.49	0.49	0.51	2.08
Tier 3	0.49	0.45	0.46	2.28

Table 4.3: Test errors MAE[ΔH] [meV] of the model on different phases.

the predictions on Pnma structures are accurate at low energies. However, there are three Pnma structures that have decomposition energies of over 700 meV, which is beyond the energy range of the other three phases. The three structures are also outliers in terms of the prediction errors that for **Tier 1** model are 46 meV, 53 meV, and 147 meV, whereas all the other errors in the whole test set remain well below 10 meV. If the three outliers are left out of the consideration, the prediction errors are 0.57 meV, 0.53 meV, and 0.55 meV for the model tiers 1, 2, and 3, respectively.

The atomic structures behind the three outliers were investigated in order to find what separates them from the rest of the test structures. The elemental content is clearly not the dividing factor because the three structures have Cl-contents of 46%, 50%, and 67%, respectively, setting them comfortably within the more populated part of the data set in that regard. Instead, the reason for the poor predictions seems to be the octahedral tilting in the structures. The three outliers are the three most heavily tilted structures in the test set. In fact, they are all within the four most heavily tilted structures in the whole **Dataset 1**. Knowing this, the large prediction errors make perfect sense. The model can only predict reliably on structures that are well represented by the training set.



(a) Pm$\bar{3}$m **(b)** P4/mbm

(c) I4/mcm **(d)** Pnma

Figure 4.7: ΔH^{DFT} vs. ΔH^{pred} plots for the four phases.

5. Structural Optimization

Structural optimization is the task of finding the atomic geometry that optimizes some quantity related to it. The most common optimization task is the minimization of the total energy. This was also the case in the structural optimization trajectories of **Dataset 2**, where DFT was used to find the local minimum energy structures starting from algorithmically sampled perovskite geometries. The ultimate test for the machine learning model is to use it in structural optimizations as a replacement for the expensive DFT calculations, while attempting to replicate the results of **Dataset 2** in a more efficient way.

The energy prediction capabilities of the ML model were already demonstrated in the previous chapter, and in principle, the model could be used for structural optimization as is. However, a more efficient approach is to differentiate the model and use the predicted gradients for the optimization. In this chapter, I go through the process of differentiating the model, testing the model gradients by predicting atomic forces in **Dataset 2** structures, and finally using the model for structural optimizations.

5.1 Model Differentiation

In order to do force calculations and structural optimization with the ML model, its derivatives with regards to the atom coordinates need to be known. I derived the model gradients mathematically starting from the KRR and MBTR definitions.

The derivations can be found in Appendix A. Then, I modified the `DScribe` and `scikit-learn` implementations of MBTR and KRR so that they calculate the gradients based on the mathematical derivations. The changes in the code do not affect model training and thus the models that were selected in Chapter 2.3 can still be used. The only difference is that now these models have an added option of predicting the energy gradients.

The gradient implementation was tested by comparing the ML model predictions to finite difference calculations. **Tier 1** model was used for predicting all 120 positional derivatives of eight random structures in the test set. The results can be seen in Figure 5.1. The mean absolute error between the direct derivatives and the finite difference estimates is approximately 1.1×10^{-3} eV/Å. The difference can arise from the gradient implementation or from the inaccuracy of the finite difference estimations. Either way, it is an upper limit for the inaccuracy of the model gradient implementation.

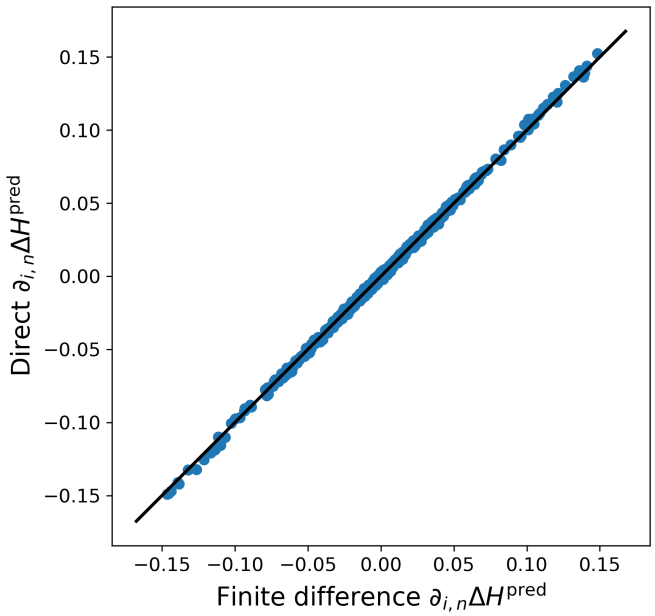


Figure 5.1: Results from the ML model gradient testing. Comparison between the energy gradient components from finite difference calculations and direct model predictions.

	$Pm\bar{3}m$	Pnma
Tier 1	18	26
Tier 2	20	26
Tier 3	22	25

Table 5.1: Force component prediction errors MAE[F_i] meV/Å on the initial structures in **Dataset 2** for the two phases and all model tiers.

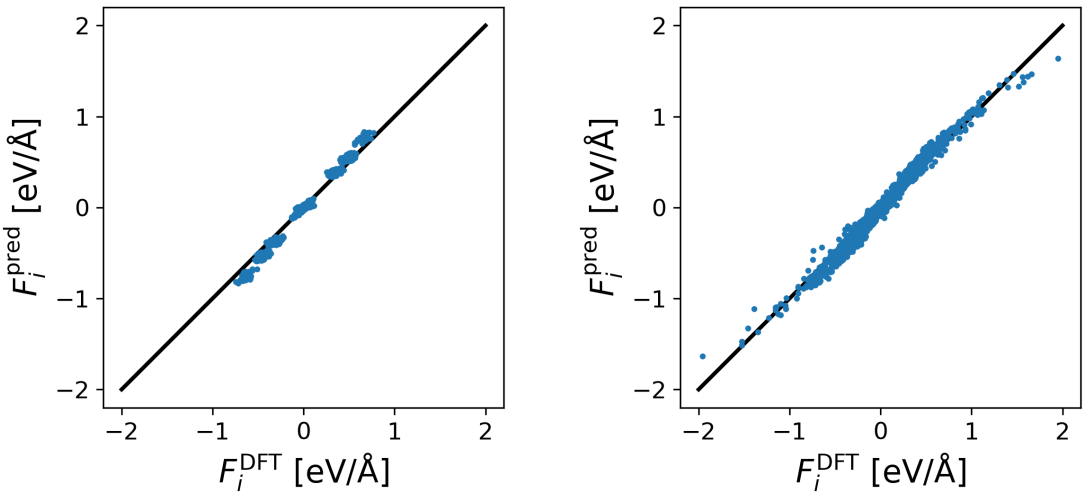
5.2 Force Predictions

Having differentiated the model, it is in principle ready to be used in structural optimization. However, before doing so, the gradient prediction accuracy was tested on atomic structures in **Dataset 2**. The atomic forces in these structures are known from DFT calculations.

I showed in Chapter 3.1.3 that there is a linear dependence between total energy gradients and ΔH gradients (see equation 3.7) and thus the ML model can be used to predict total energy gradients and atomic forces:

$$F_{n,i}^{\text{pred}} = -8\partial_{n,i}\Delta H^{\text{pred}}. \tag{5.1}$$

There are 100 structural optimization trajectories in **Dataset 2**. The initial structures come from **Dataset 1** but were not included in the training set when the three differently tiered models were fitted in Chapter 4.2, allowing for unbiased test results. First, the peak performances of the three models were evaluated by looking only at the initial structures. The 120 force components of all 100 structures were predicted with the three models and compared to the forces obtained from the DFT calculations. The mean absolute errors of force component predictions are shown in Table 5.1. With all models, the prediction error is about 20 meV/Å for $Pm\bar{3}m$ and 25 meV/Å for Pnma. The model accuracy does not increase with the tier. In fact, **Tier 1** model performs somewhat better than the other two. Knowing that

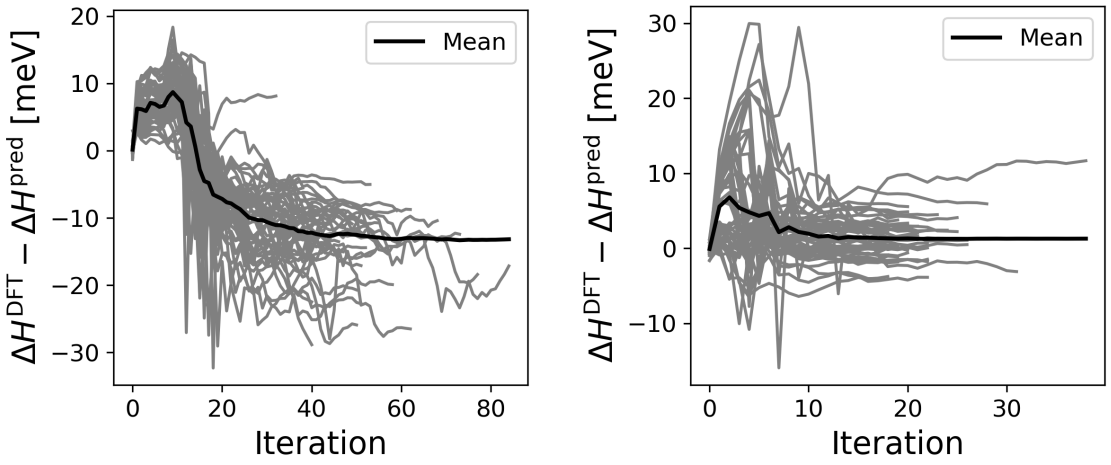


(a) Pm$\bar{3}$m **(b)** Pnma

Figure 5.2: DFT force components of the initial structures in **Dataset 2** compared to the corresponding **Tier 1** predictions.

Tier 1 is also much faster than the other models, it was picked as the model that is used for the rest of the predictions and structural optimizations. Its predictions have been visualized in Figure 5.2.

Next, **Tier 1** model was used for making predictions on all of **Dataset 2**. The predicted decomposition energies and forces were compared to the DFT calculations. In Figure 5.3 the energy differences between the predicted values and DFT have been plotted against the optimization iteration step. At low iterations the differences are very close to zero, but when the optimizations proceed the prediction errors increase immediately. The plots also show that the ML model starts to systematically overestimate the energies of Pm$\bar{3}$m structures. The difference between the phases shows in the mean absolute errors on the final optimized structures, which are 13.5 meV and 2.3 meV for Pm$\bar{3}$m and Pnma, respectively. The force prediction results are shown in Figure 5.4. Similarly to the energy predictions, the force accuracy decreases with the iterations. The mean absolute force component errors for



(a) $Pm\bar{3}m$ **(b)** Pnma

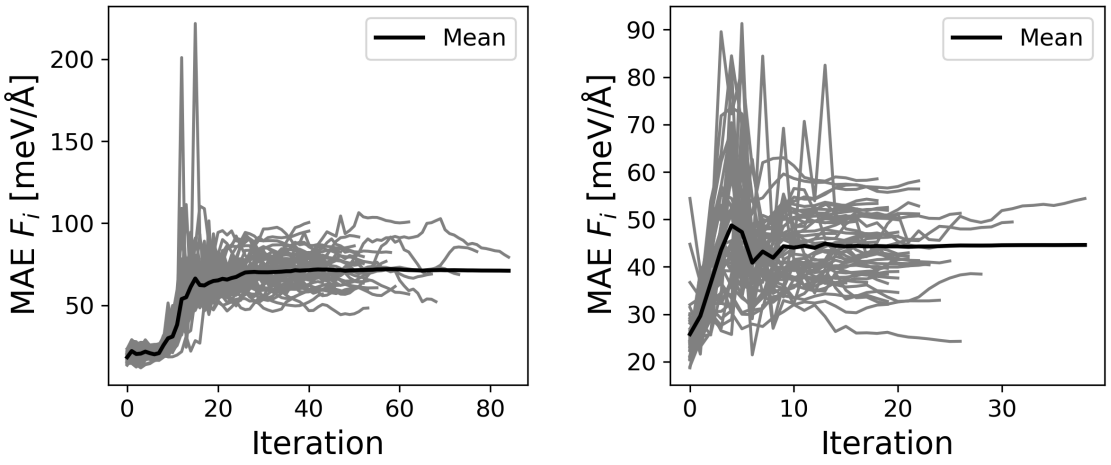
Figure 5.3: The difference between DFT energies and **Tier 1** predictions with regards to the iteration step in the structural optimization trajectories of **Dataset 2**.

the final optimized structures are 71 meV/Å and 45 meV/Å, with Pnma predictions being more accurate in this category as well.

5.3 Structural Optimization

The final test for the model was to use its force predictions in an attempt to replicate DFT optimization results. If successful, this would allow for a considerable speed up in finding minimum energy perovskite structures. The DFT optimization trajectories of **Dataset 2** were generated in `FHI-aims` by minimizing the total energy in terms of the atom positions using the BFGS algorithm [52]. This can be imitated by combining the ML model with BFGS, replacing the slow DFT computations with the force predictions of the model. BFGS implementation identical to the one in `FHI-aims` could not be found. In the end I chose to use `ASE` (Atomic Simulation Environment) [53] and its implementation of BFGS.

The ML optimization scheme was used on the same 100 initial structures that

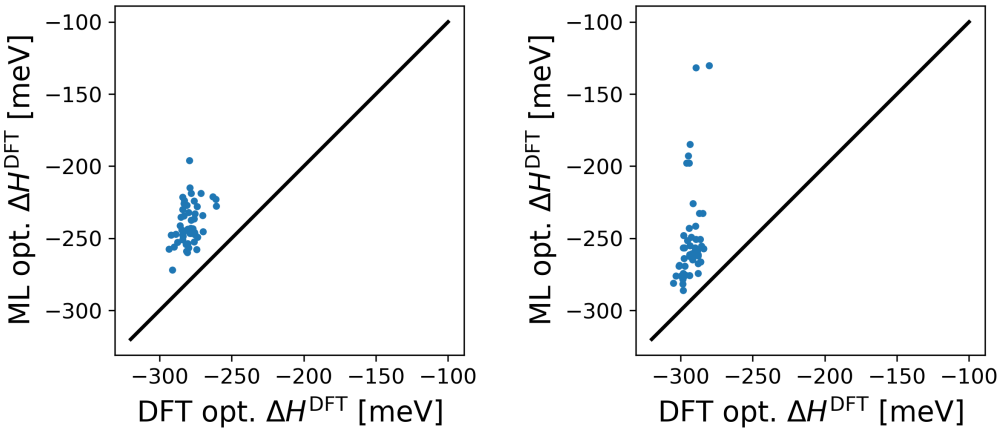


(a) Pm$\bar{3}$m **(b)** Pnma

Figure 5.4: The mean absolute force component errors of **Tier 1** predictions with regards to the iteration step in the structural optimization trajectories of **Dataset 2**.

were the starting points for the DFT optimizations in **Dataset 2**. The optimizations were stopped when the maximum predicted force got below 5×10^{-3} eV/Å. This stopping criterion is stricter than the one used in the DFT optimizations, where the same convergence limit value was used but mean forces were considered instead of maximum forces.

After the ML optimizations were done, the total energies of the resulting optimum structures were calculated with DFT using the same settings as in the generation of the data sets. The final energies form the two optimization methods are compared in Figure 5.5. The structures found in ML optimization have decomposition energies that are systematically higher than their DFT counterparts. The differences are 39 meV and 42 meV on average, which is much more than the standard deviation in the target energies. This means that the resolution of the ML optimizations is not high enough to make a distinction between the different structures. However, the ML optimization did manage to decrease the energy of the structures. The mean energy reductions were 78 meV and 106 meV for Pm$\bar{3}$m and

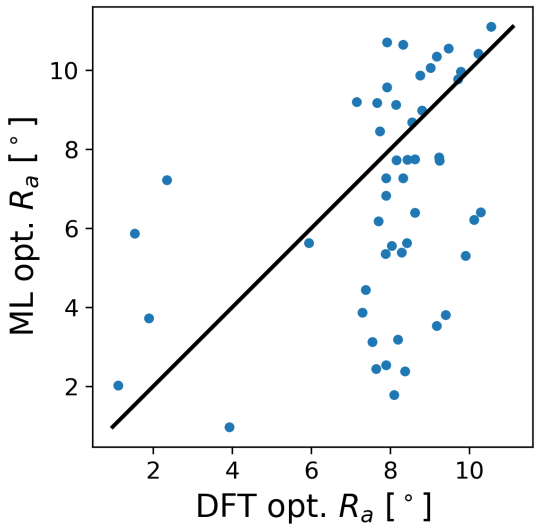


(a) Pm$\bar{3}$m **(b)** Pnma

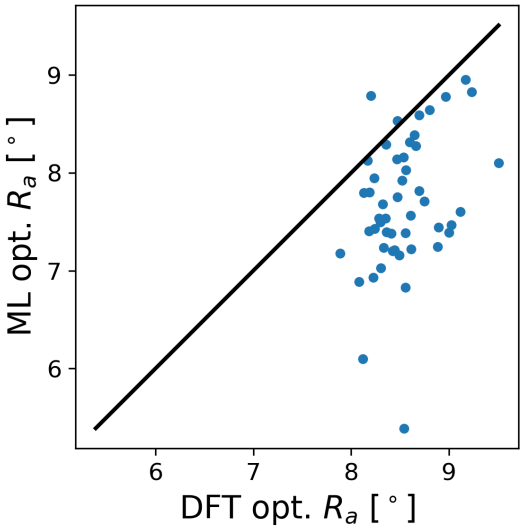
Figure 5.5: Comparison of the DFT calculated decomposition energies ΔH^{DFT} of the final structures obtained with the DFT and ML optimization schemes.

Pnma, respectively, compared 117 meV and 148 meV that were achieved with DFT optimizations.

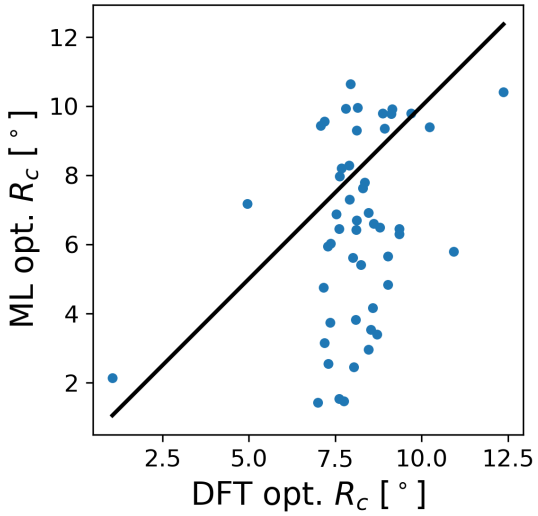
Next, I compared the ML-optimized and DFT-optimized atomic geometries in terms of typical structural parameters of perovskites. The structures were characterized in terms of octahedral rotation angles that were estimated in the way that was introduced in Chapter 3. The results of the comparison are shown in Figure 5.6. The mean absolute errors are 2.4° and 0.8° for Pm$\bar{3}$m and Pnma, respectively, but as was the case with energies of the optimized structures, the standard deviation in the angles of the target structures is smaller than the prediction errors. The ML optimization has taken the structures closer to the correct optima in terms of octahedral rotation angles as well. The ML-optimized Pm$\bar{3}$ structures are 5.8° closer to the DFT optima than the initial structures were. For Pnma structures the mean improvement is 3.2°. The ML model does make some systematic errors in the optimization. This is especially the case in Pnma structures where a and b-directional rotation angles are underestimated in all but two cases.



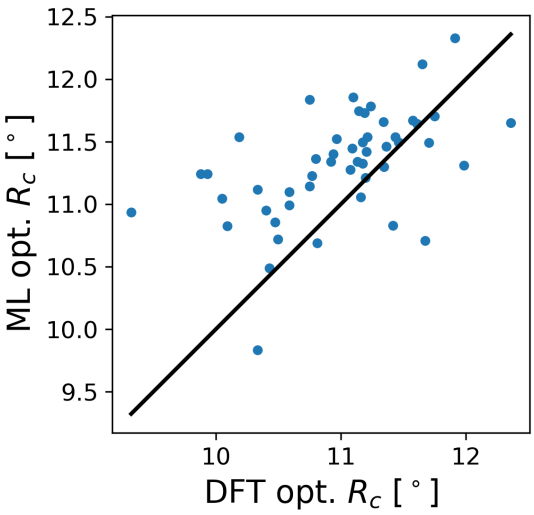
(a) Pm$\bar{3}$m, R_a



(b) Pnma, R_a



(c) Pm$\bar{3}$m, R_c



(d) Pnma, R_c

Figure 5.6: Comparison of the octahedral rotation angles R_a and R_c of the final structures obtained with the DFT and ML optimization schemes.

6. Summary

6.1 Discussion

In this section, I revisit the key results of the thesis and discuss their meaning and impact. In Chapter 4, the machine learning model was fitted to **Dataset 1**. A two-level Bayesian optimization search was used for optimizing the hyperparameters of the model. The method was successful at finding optimal values for parameters σ, α, and γ efficiently due to the separation of MBTR vector calculation and KRR parameter optimization. I used the method to select three models with different cutoff radii of contributing atom pairs in the structure descriptor. All three models performed similarly in decomposition energy prediction. When the three Pnma outlier structures were not considered, the prediction errors for all four phases were under 0.6 meV. This corresponds to an impressive 0.12 meV prediction error per atom and allows accurate predictions of the convex hull for alloy exploration. Predicting the decomposition energy of an atomic geometry takes under a hundred milliseconds on one CPU core. The time save compared to DFT is over five orders of magnitude.

The machine learning model was differentiated in Chapter 5. The numerical accuracy of the gradient implementation is difficult to assess precisely, but I found it to be under 1.1 meV/Å, which corresponds to $8 \times 1.1\,\text{meV/Å} = 8.8\,\text{meV/Å}$ in force predictions. The mean force prediction errors on **Dataset 2** structures were between 20 meV/Å and 71 meV/Å, exceeding the numerical inaccuracy considerably.

This indicates that the numerical accuracy of the gradient implementation was not the primary source of error in force predictions and structural optimization.

When the model was tested on the DFT structural optimization trajectories of **Dataset 2**, I observed that the prediction errors in both energy and force increased when the optimization progressed. The mean prediction errors for decomposition energy and forces went from 0.56 meV and 18 meV/Å to 14 meV and 71 meV/Å for the $Pm\bar{3}m$ structures. For the Pnma structures, the errors went from 0.49 meV and 25 meV/Å to 2.3 meV and 45 meV/Å. This loss in accuracy can easily be explained. The initial structures were from **Dataset 1** and thus the training data of the model consists of structures that are similar to them. When the structures start to change during the optimization, they do not correspond as well to the training data anymore. This is especially true for $Pm\bar{3}m$ structures that initially have cubic symmetry but start to exhibit octahedral tilting as a result of the optimization. The training data does not include atomic structures that have cubic simulation cells and octahedral tilting. The fact that the optimized structures do not resemble the training structures causes the model to extrapolate, which decreases the prediction accuracy.

At the end of Chapter 5, the ML model was used for structural optimization in an attempt to replicate the DFT optimization results of **Dataset 2**. The results were replicated qualitatively. Just like in the DFT optimizations, octahedral rotations were introduced to the $Pm\bar{3}m$ structures. The distribution of octahedral rotation angles in Pnma structures got narrower when they were relaxed, as was the case in the DFT relaxations. The ML optimization reduced the decomposition energy of structures and brought the geometries closer to the correct optima in terms of octahedral rotation angles.

The ML optimization results did not match DFT results quantitatively. The decomposition energies of the ML-optimized $Pm\bar{3}m$ structures were on average 39 meV above the correct optimum. For the Pnma structures the difference was

42 meV. The energy resolution of the ML optimizations is likely insufficient for distinguishing between the structures that have the lowest energy. The ML model needs to be improved before it can be applied on the optimization of $CsPb(Cl, Br)_3$ stability over the substitution range. The most obvious explanation for the observed structural optimization results is extrapolation, which was already observed with the predictions on **Dataset 2** trajectories. However, it is somewhat surprising that the optimization results were so similar for both $Pm\bar{3}m$ and Pnma, with the decomposition energy being overestimated by about 40 meV. In the energy and force predictions on **Dataset 2**, the errors on the Pnma structures were much smaller, this does not seem to affect the ML optimization results.

In this thesis I was presented with a fixed dataset. New training data could not be generated before the end of the project. Future work (see Section 6.2) will therefore need to take the lessons learned in this project into account when generating improved training data.

Even thought the desired accuracy for structure optimization was not quite reached during the course of my Masters project, a similar machine learning model could have many uses outside of this work. The model performed very well in property prediction, being orders of magnitude faster than DFT while predicting decomposition energies with high accuracy. This excellent accuracy-speed ratio would be very useful in any study where the number of atomic structures is too large to handle with DFT. Additionally, the ML optimizations did reduce the energy of the structures, which means that the model could be used to accelerate structural optimization, by initiating the process with a model-driven relaxation and only finishing with DFT. This approach would decrease the number of DFT calculations required to find the correct optimum, and thus save computational resources. In this work, I took the global structure representation approach to atomic force prediction instead of the more commonly used local approach used in force fields. The competitiveness

of this new alternative approach can be assessed in the future after the approach has been improved.

6.2 Future Work

There are several ways to improve the current model performance. The first way would be to modify the training data. In the tests, the model predictions showed clear signs of extrapolation. In order to fix this problem, the training data would have to be expanded. Some of the optimization trajectories of **Dataset 2**, or similar to them, could be added to the training set. Alternatively, the structure sampling method that was used to create **Dataset 1** could be changed so that a wider coverage of different structures was achieved. The drawback of extending the training set is that with KRR the computational costs of prediction and training are directly proportional to the training set size. Furthermore, the more training data there is, the more DFT calculations are needed to get the labels for the data. Managing the training set size is a balancing act between computational efficiency and model accuracy.

There is another reason to revise the training data sampling method. Due to the random assignment of each X-site atom element, the resulting data set lacked structures with high and low Cl concentrations. In order to scan the whole substitution range, all concentrations should be present in the training set. The solution is to sample the concentrations of structures from a distribution that guarantees that the ends of the substitution range are present, and only permute the Cl and Br atoms randomly.

Another possible improvement to the model would be to replace the KRR component with a Gaussian process regression (GPR) model [54]. GPR is very similar to KRR, but provides uncertainty estimates on its predictions. The uncertainties would tell which atomic structures are not covered by the training data, and could

help at improving the training data set. The drawback of GPR is that it is more difficult to differentiate.

If the improvements to the model were successful, the next task would be to apply the model in solving more complex perovskite design problems. Starting from the optimization of stability for $CsPb(Cl, Br)_3$, to other properties and more complicated perovskite structures. The ultimate objective would be to solve the design problem of stability, intoxicity, and photovoltaic efficiency of $MAPbI_3$.

6.3 Conclusions

The thesis presents new machine learning methodology for designing perovskite materials for solar energy applications. The approach targeted accelerated property prediction and atomic structure optimization for halide perovskites. The machine learning model used the MBTR to represent atomic structures in vector form and the KRR machine learning approach to map the vectors to the material property. The methodology was tested by training the model to predict decomposition energies of $CsPb(Cl, Br)_3$ composite perovskite structures. I also differentiated the model and tested its ability to reproduce results from DFT structural optimization.

The machine learning model was capable of predicting the decomposition energies of perovskite structures with an accuracy of close to 0.1 meV per atom. The time saved over DFT was approximately five orders of magnitude. The structural optimization results were promising but improvements in the model training are needed to enhance the accuracy of machine-learning-driven optimization to a level that is required in solving complex material design problems. In conclusion, the approach has potential to be effective in the discovery of new improved perovskite materials.

References

1 Ottmar Edenhofer. *Climate change 2014: mitigation of climate change*, volume 3. Cambridge University Press, 2015.

2 Dolf Gielen, Francisco Boshell, Deger Saygin, Morgan D Bazilian, Nicholas Wagner, and Ricardo Gorini. The role of renewable energy in the global energy transformation. *Energy Strategy Reviews*, 24:38–50, 2019.

3 Kevin A Bush, Salman Manzoor, Kyle Frohna, Zhengshan J Yu, James A Raiford, Axel F Palmstrom, Hsin-Ping Wang, Rohit Prasanna, Stacey F Bent, Zachary C Holman, et al. Minimizing current and voltage losses to reach 25% efficient monolithic two-terminal perovskite–silicon tandem solar cells. *ACS Energy Letters*, 3(9):2173–2180, 2018.

4 Rui Wang, Muhammad Mujahid, Yu Duan, Zhao-Kui Wang, Jingjing Xue, and Yang Yang. A review of perovskites solar cell stability. *Advanced Functional Materials*, 29(47):1808843, 2019.

5 J Schoonman. Organic–inorganic lead halide perovskite solar cell materials: a possible stability problem. *Chemical Physics Letters*, 619:193–195, 2015.

6 Kenjiro Miyano, Masatoshi Yanagida, Neeti Tripathi, and Yasuhiro Shirai. Hysteresis, stability, and ion migration in lead halide perovskite photovoltaics. *The journal of physical chemistry letters*, 7(12):2240–2245, 2016.

7 Giles E Eperon, Tomas Leijtens, Kevin A Bush, Rohit Prasanna, Thomas Green, Jacob Tse-Wei Wang, David P McMeekin, George

Volonakis, Rebecca L Milot, Richard May, et al. Perovskite-perovskite tandem photovoltaics with optimized band gaps. *Science*, 354(6314):861–865, 2016.

8 Meysam Pazoki and Tomas Edvinsson. Metal replacement in perovskite solar cell materials: chemical bonding effects and optoelectronic properties. *Sustainable Energy & Fuels*, 2(7):1430–1445, 2018.

9 `https://www.nrel.gov/pv/cell-efficiency.html`.

10 AM Glazer. The classification of tilted octahedra in perovskites. *Acta Crystallographica Section B: Structural Crystallography and Crystal Chemistry*, 28(11):3384–3392, 1972.

11 Peng Gao, Michael Grätzel, and Mohammad K Nazeeruddin. Organohalide lead perovskites for photovoltaic applications. *Energy & Environmental Science*, 7(8):2448–2463, 2014.

12 Yong Wang, M Ibrahim Dar, Luis K Ono, Taiyang Zhang, Miao Kan, Yawen Li, Lijun Zhang, Xingtao Wang, Yingguo Yang, Xingyu Gao, et al. Thermodynamically stabilized β-cspbi3–based perovskite solar cells with efficiencies> 18%. *Science*, 365(6453):591–595, 2019.

13 David A Egger, Achintya Bera, David Cahen, Gary Hodes, Thomas Kirchartz, Leeor Kronik, Robert Lovrincic, Andrew M Rappe, David R Reichman, and Omer Yaffe. What remains unexplained about the properties of halide perovskites? *Advanced Materials*, 30(20):1800691, 2018.

14 Valerio D'innocenzo, Giulia Grancini, Marcelo JP Alcocer, Ajay Ram Srimath Kandada, Samuel D Stranks, Michael M Lee, Guglielmo Lanzani, Henry J Snaith, and Annamaria Petrozza. Excitons versus free

charges in organo-lead tri-halide perovskites. *Nature communications*, 5(1):1–6, 2014.

15 Qingfeng Dong, Yanjun Fang, Yuchuan Shao, Padhraic Mulligan, Jie Qiu, Lei Cao, and Jinsong Huang. Electron-hole diffusion lengths> 175 μm in solution-grown ch3nh3pbi3 single crystals. *Science*, 347(6225):967–970, 2015.

16 Yu Bi, Eline M Hutter, Yanjun Fang, Qingfeng Dong, Jinsong Huang, and Tom J Savenije. Charge carrier lifetimes exceeding 15 μs in methylammonium lead iodide single crystals. *The journal of physical chemistry letters*, 7(5):923–928, 2016.

17 Guichuan Xing, Nripan Mathews, Shuangyong Sun, Swee Sien Lim, Yeng Ming Lam, Michael Grätzel, Subodh Mhaisalkar, and Tze Chien Sum. Long-range balanced electron-and hole-transport lengths in organic-inorganic ch3nh3pbi3. *Science*, 342(6156):344–347, 2013.

18 Bo Chen, Peter N Rudd, Shuang Yang, Yongbo Yuan, and Jinsong Huang. Imperfections and their passivation in halide perovskite solar cells. *Chemical Society Reviews*, 48(14):3842–3867, 2019.

19 Weiqiang Liao, Dewei Zhao, Yue Yu, Niraj Shrestha, Kiran Ghimire, Corey R Grice, Changlei Wang, Yuqing Xiao, Alexander J Cimaroli, Randy J Ellingson, et al. Fabrication of efficient low-bandgap perovskite solar cells by combining formamidinium tin iodide with methylammonium lead iodide. *Journal of the American Chemical Society*, 138(38):12360–12363, 2016.

20 Hairen Tan, Ankit Jain, Oleksandr Voznyy, Xinzheng Lan, F Pelayo García De Arquer, James Z Fan, Rafael Quintero-Bermudez,

Mingjian Yuan, Bo Zhang, Yicheng Zhao, et al. Efficient and stable solution-processed planar perovskite solar cells via contact passivation. *Science*, 355(6326):722–726, 2017.

21 Chong Liu, Wenzhe Li, Cuiling Zhang, Yunping Ma, Jiandong Fan, and Yaohua Mai. All-inorganic cspbi2br perovskite solar cells with high efficiency exceeding 13%. *Journal of the American chemical society*, 140(11):3825–3828, 2018.

22 Soumyo Chatterjee and Amlan J Pal. Influence of metal substitution on hybrid halide perovskites: towards lead-free perovskite solar cells. *Journal of Materials Chemistry A*, 6(9):3793–3823, 2018.

23 Sneha A Kulkarni, Tom Baikie, Pablo P Boix, Natalia Yantara, Nripan Mathews, and Subodh Mhaisalkar. Band-gap tuning of lead halide perovskites using a sequential deposition process. *Journal of Materials Chemistry A*, 2(24):9221–9225, 2014.

24 Jingrui Li, Jari Järvi, and Patrick Rinke. Multiscale model for disordered hybrid perovskites: The concept of organic cation pair modes. *Physical Review B*, 98(4):045201, 2018.

25 Haoyan Huo and Matthias Rupp. Unified representation for machine learning of molecules and crystals. *arXiv preprint arXiv:1704.06439*, 13754, 2017.

26 Katja Hansen, Grégoire Montavon, Franziska Biegler, Siamac Fazli, Matthias Rupp, Matthias Scheffler, O Anatole Von Lilienfeld, Alexandre Tkatchenko, and Klaus-Robert Muller. Assessment and validation of machine learning methods for predicting molecular atomization en-

ergies. *Journal of Chemical Theory and Computation*, 9(8):3404–3419, 2013.

27 Pierre Hohenberg and Walter Kohn. Inhomogeneous electron gas. *Physical review*, 136(3B):B864, 1964.

28 Robert G Parr. Density functional theory of atoms and molecules. In *Horizons of quantum chemistry*, pages 5–15. Springer, 1980.

29 Richard M Martin. *Electronic structure: basic theory and practical methods*. Cambridge university press, 2020.

30 Carlos Fiolhais, Fernando Nogueira, and Miguel AL Marques. *A primer in density functional theory*, volume 620. Springer Science & Business Media, 2003.

31 Robert O Jones. Density functional theory: Its origins, rise to prominence, and future. *Reviews of modern physics*, 87(3):897, 2015.

32 Kieron Burke. Perspective on density functional theory. *The Journal of chemical physics*, 136(15):150901, 2012.

33 Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Physical review*, 140(4A):A1133, 1965.

34 John P Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Physical review letters*, 77(18):3865, 1996.

35 Volker Blum, Ralf Gehrke, Felix Hanke, Paula Havu, Ville Havu, Xinguo Ren, Karsten Reuter, and Matthias Scheffler. Ab initio molecular simulations with numeric atom-centered orbitals. *Computer Physics Communications*, 180(11):2175–2196, 2009.

36 Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

37 Tom M Mitchell et al. Machine learning. 1997.

38 Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole Von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, 108(5):058301, 2012.

39 Joohwi Lee, Atsuto Seko, Kazuki Shitara, Keita Nakayama, and Isao Tanaka. Prediction model of band gap for inorganic compounds by combination of density functional theory calculations and machine learning techniques. *Physical Review B*, 93(11):115104, 2016.

40 Ya Zhuo, Aria Mansouri Tehrani, and Jakoah Brgoch. Predicting the band gaps of inorganic solids by machine learning. *The journal of physical chemistry letters*, 9(7):1668–1673, 2018.

41 Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

42 Albert P Bartók, Mike C Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical review letters*, 104(13):136403, 2010.

43 Felix A Faber, Luke Hutchison, Bing Huang, Justin Gilmer, Samuel S Schoenholz, George E Dahl, Oriol Vinyals, Steven Kearnes, Patrick F Riley, and O Anatole Von Lilienfeld. Prediction errors of molecular machine learning models lower than hybrid dft error. *Journal of chemical theory and computation*, 13(11):5255–5264, 2017.

44 Annika Stuke, Milica Todorović, Matthias Rupp, Christian Kunkel, Kunal Ghosh, Lauri Himanen, and Patrick Rinke. Chemical diversity in molecular orbital energy predictions with kernel ridge regression. *The Journal of chemical physics*, 150(20):204121, 2019.

45 Lauri Himanen, Marc OJ Jäger, Eiaki V Morooka, Filippo Federici Canova, Yashasvi S Ranawat, David Z Gao, Patrick Rinke, and Adam S Foster. Dscribe: Library of descriptors for machine learning in materials science. *Computer Physics Communications*, 247:106949, 2020.

46 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

47 Craig Saunders, Alexander Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. 1998.

48 Jingrui Li and Patrick Rinke. Atomic structure of metal-halide perovskites from first principles: The chicken-and-egg paradox of the organic-inorganic interaction. *Physical Review B*, 94(4):045201, 2016.

49 John P Perdew, Adrienn Ruzsinszky, Gábor I Csonka, Oleg A Vydrov, Gustavo E Scuseria, Lucian A Constantin, Xiaolan Zhou, and Kieron Burke. Restoring the density-gradient expansion for exchange in solids and surfaces. *Physical review letters*, 100(13):136406, 2008.

50 Milica Todorović, Michael U Gutmann, Jukka Corander, and Patrick Rinke. Bayesian inference of atomistic structure in functional materials. *Npj computational materials*, 5(1):1–7, 2019.

51 Boss. `https://gitlab.com/cest-group/boss`.

52 Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

53 Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, Eric D Hermes, Paul C Jennings, Peter Bjerre Jensen, James Kermode, John R Kitchin, Esben Leonhard Kolsbjerg, Joseph Kubal, Kristen Kaasbjerg, Steen Lysgaard, Jón Bergmann Maronsson, Tristan Maxson, Thomas Olsen, Lars Pastewka, Andrew Peterson, Carsten Rostgaard, Jakob Schiøtz, Ole Schütt, Mikkel Strange, Kristian S Thygesen, Tejs Vegge, Lasse Vilhelmsen, Michael Walter, Zhenhua Zeng, and Karsten W Jacobsen. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, 2017.

54 Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

A. Model Gradient Derivation

The decomposition energy prediction given by the model was defined in equation (2.15). The derivative of the prediction with regards to the ith coordinate of the nth atom in structure $\boldsymbol{s}$ is

$$\begin{aligned}\frac{\partial}{\partial(\boldsymbol{R}_n)_i}\Delta H^{\text{pred}}(\boldsymbol{s}) \equiv& \partial_{n,i}\Delta H^{\text{pred}}(\boldsymbol{s}) \\ =& \partial_{n,i}\sum_j^N \beta_j k(\boldsymbol{s},\boldsymbol{s}_j) \\ =& \sum_j^N \beta_j \partial_{n,i} k(\boldsymbol{s},\boldsymbol{s}_j),\end{aligned} \tag{A.1}$$

where the kernel derivative can be found by applying chain rule on equation (2.16)

$$\begin{aligned}\partial_{n,i}k(\boldsymbol{s},\boldsymbol{s}_j) =& \partial_{n,i} e^{-\gamma||\boldsymbol{M}(\boldsymbol{s})-\boldsymbol{M}(\boldsymbol{s}_j)||_2^2} \\ =& e^{-\gamma||\boldsymbol{M}(\boldsymbol{s})-\boldsymbol{M}(\boldsymbol{s}_j)||_2^2}\partial_{n,i}\left(-\gamma||\boldsymbol{M}(\boldsymbol{s})-\boldsymbol{M}(\boldsymbol{s}_j)||_2^2\right) \\ =& k(\boldsymbol{s},\boldsymbol{s}_j)\left(-\gamma\partial_{n,i}||\boldsymbol{M}(\boldsymbol{s})-\boldsymbol{M}(\boldsymbol{s}_j)||_2^2\right) \\ =& -\gamma k(\boldsymbol{s},\boldsymbol{s}_j)\left(\boldsymbol{M}(\boldsymbol{s})-\boldsymbol{M}(\boldsymbol{s}_j)\right)\cdot\partial_{n,i}\boldsymbol{M}(\boldsymbol{s}).\end{aligned} \tag{A.2}$$

As can be seen from equation (A.2), the corresponding derivative needs to be calculated for the MBTR representation. This can be done in small parts. Starting from function g that is defined in equation (2.9).

$$\begin{aligned}\partial_{n,i}g^{l,m} =& \partial_{n,i}\frac{1}{|\boldsymbol{R}_l-\boldsymbol{R}_m|} \\ =& -(\delta_{nl}+\delta_{nm})\frac{(\boldsymbol{R}_l-\boldsymbol{R}_m)_i}{|\boldsymbol{R}_l-\boldsymbol{R}_m|^3} \\ =& -(\delta_{nl}+\delta_{nm})\left(g^{l,m}\right)^3(\boldsymbol{R}_l-\boldsymbol{R}_m)_i\end{aligned} \tag{A.3}$$

Similarly for the weight function from equation (2.6):

$$\begin{aligned}
\partial_{n,i} w^{l,m} =& \partial_{n,i} e^{-s|\boldsymbol{R}_l - \boldsymbol{R}_m|} \\
=& - e^{-s|\boldsymbol{R}_l - \boldsymbol{R}_m|} s \partial_{n,i} |\boldsymbol{R}_l - \boldsymbol{R}_m| \\
=& - (\delta_{nl} + \delta_{nm}) w^{l,m} s \frac{(\boldsymbol{R}_l - \boldsymbol{R}_m)_i}{|\boldsymbol{R}_l - \boldsymbol{R}_m|} \\
=& w^{l,m} \frac{1}{\left(g^{l,m}\right)^2} \partial_{n,i} g^{l,m}.
\end{aligned} \tag{A.4}$$

The derivative of $d^{l,m}$ from equation (2.8) is

$$\begin{aligned}
\partial_{n,i} d^{l,m}(x) =& \partial_{n,i} \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-g^{l,m})^2}{2\sigma^2}} \right) \\
=& \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-g^{l,m})^2}{2\sigma^2}} \partial_{n,i} \left(-\frac{(x-g^{l,m})^2}{2\sigma^2} \right) \\
=& d^{l,m}(x) \frac{1}{\sigma^2} (x - g^{l,m}) \partial_{n,i} g^{l,m}.
\end{aligned} \tag{A.5}$$

Knowing the derivatives for bot $w^{l,m}$ and $d^{l,m}$, the partial MBTR function from equation (2.5) can now be differentiated.

$$\begin{aligned}
\partial_{n,i} \mathrm{MBTR}^{Z_1,Z_2}(x) =& \partial_{n,i} \sum_{l}^{|Z_1|} \sum_{m}^{|Z_2|} w^{l,m} d^{l,m}(x) \\
=& \sum_{l}^{|Z_1|} \sum_{m}^{|Z_2|} \left[d^{l,m}(x) \partial_{n,i} w^{l,m} + w^{l,m} \partial_{n,i} d^{l,m}(x) \right] \\
=& \sum_{l}^{|Z_1|} \sum_{m}^{|Z_2|} \left[d^{l,m}(x) w^{l,m} \frac{1}{\left(g^{l,m}\right)^2} \partial_{n,i} g^{l,m} + w^{l,m} d^{l,m}(x) \frac{1}{\sigma^2} (x - g^{l,m}) \partial_{n,i} g^{l,m} \right] \\
=& \sum_{l}^{|Z_1|} \sum_{m}^{|Z_2|} w^{l,m} d^{l,m}(x) \left[\frac{1}{\left(g^{l,m}\right)^2} + \frac{1}{\sigma^2} \left(x - g^{l,m} \right) \right] \partial_{n,i} g^{l,m}.
\end{aligned} \tag{A.6}$$

The discretization of the MBTR derivatives needs to be treated carefully. As can be seen from (A.6), there are now two differently shaped distributions: $d^{l,m}(x)$ and $xd^{l,m}(x)$. In order for the derivatives to correspond correctly to the discretized MBTR vectors, both distributions need to be estimated through their cumulative distributions. This was already done for the $d^{l,m}(x)$ distribution when calculating the representation itself. The second distribution can be given its own name for the

sake of convenience: $c^{l,m}(x) := xd^{l,m}(x)$. Its cumulative distribution is

$$\begin{aligned} C^{l,m}(x) =& \int_{-\infty}^{x} x' d^{l,m}(x') dx' \\ =& \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} x' e^{-\frac{(x'-g^{l,m})^2}{2\sigma^2}} dx' \\ =& g^{l,m} \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{x - g^{l,m}}{\sigma\sqrt{2}} \right) \right] - \frac{\sigma}{\sqrt{2\pi}} \left[e^{-\frac{(x-g^{l,m})^2}{2\sigma^2}} - 1 \right] \\ =& g^{l,m} D^{l,m}(x) - \frac{\sigma}{\sqrt{2\pi}} \left[e^{-\frac{(x-g^{l,m})^2}{2\sigma^2}} - 1 \right] \end{aligned} \tag{A.7}$$

$C^{l,m}(x)$ can be used to estimate $c^{l,m}(x)$ at discrete $x-$values similarly to (2.13) in order to obtain vectors $\boldsymbol{c}^{l,m}$. Then the two different distributions in (A.6) should be separated:

$$\partial_{n,i}\mathrm{MBTR}^{Z_1,Z_2}(x) = \sum_{l}^{|Z_1|} \sum_{m}^{|Z_2|} w^{l,m} \left[d^{l,m}(x) \left(\frac{1}{\left(g^{l,m}\right)^2} - \frac{g^{l,m}}{\sigma^2} \right) + c^{l,m}(x) \frac{1}{\sigma^2} \right] \partial_{n,i} g^{l,m}, \tag{A.8}$$

which leads to the discretized form

$$\partial_{n,i}\boldsymbol{M}^{Z_1,Z_2} = \sum_{l}^{|Z_1|} \sum_{m}^{|Z_2|} w^{l,m} \left[\boldsymbol{d}^{l,m} \left(\frac{1}{\left(g^{l,m}\right)^2} - \frac{g^{l,m}}{\sigma^2} \right) + \boldsymbol{c}^{l,m} \frac{1}{\sigma^2} \right] \partial_{n,i} g^{l,m}. \tag{A.9}$$

The derivative of the whole MBTR vector $\boldsymbol{M}(\boldsymbol{s})$ is obtained by stacking the different elemental contributions after each other.